Integrative Modellierung kooperativer Informationssysteme

– Ein Konzept auf der Basis von Ontologien und Petri-Netzen –

Dissertation

zur Erlangung des Grades eines
Doktors der Wirtschaftswissenschaften
(Dr. rer. pol.)

durch den Fachbereich Wirtschaftswissenschaften der
Universität Duisburg-Essen
Campus Essen

vorgelegt von

Dipl.-Kfm. Yilmaz Alan
aus Krefeld

Tag der mündlichen Prüfung: 12. September 2005

Gutachter:
Univ.-Prof. Dr. St. Zelewski
Univ.-Prof. Dr. H. H. Adelsberger
Univ.-Prof. Dr. L. Mochty

Inhaltsverzeichnis

Abkürzungs- und Akronymverzeichnis

ACM .... Association for Computing Machinery

AI .... Artificial Intelligence

Aufl. .... Auflage

Bd. .... Band

CAiSE .... Conference on Advanced Information Systems Engineering

CoopIS .... Conference on Cooperative Information Systems

DAML .... Darpa Agent Markup Language

DEXA .... Database and Expert Systems Applications (Konferenz)

DL .... Description Logics

eng .... englisch

et al. .... und andere

f. .... folgende (Seite)

ff. .... folgende (Seiten)

fr .... französisch

FQAS .... Formal Query Answering Systems (Konferenz)

F-Logic .... Frame-Logic (Programmiersprache)

Fn. .... Fußnote

ger .... deutsch

Hrsg. .... Herausgeber

HTML .... Hypertext Markup Language

i.d.F. .... in diesem Fall

i.e.S. .... im engeren Sinne

i.w.S. .... im weiteren Sinne

IEEE .... Intitute of Electrical and Electronic Engineers

IFCIS .... International Foundation on Cooperative Information Systems

Jg. .... Jahrgang

KBS .... Knowledge Based System

KRAFT .... Knowledge Reuse & Fusion/Transformation (Projekt)

MAS .... Multi-Agenten-System

MASCOTS .... Modeling, Analysis, and Simulation On Computer and Telecommunication Systems (Workshop)

Nr. .... Nummer

NR/T .... Nested Relation/Transition

o.a. .... oben angeführt

o.ä. .... oder ähnlich

o.Hrsg. .... ohne Herausgeber

o.O. .... ohne Ort

o.V. .... ohne Verfasser

Oil .... Ontology Inference Layer / Ontology Interchange Language

OKBC .... Open Knowledge Base Connectivity

Pr/T-Netz .... Prädikat/Transition-Netz

PROLOG .... Programming in Logic (Programmiersprache)

RDF(S) .... Resource Description Framework (Schema)

RFC .... Request für Comments

S. .... Seite

S/T-Netz .... Stelle/Transition-Netz

SIGACT............. Special Interest Group on Algorithms and Computation Theory

SIGART............. Special Interest Group on Artificial Intelligence

SIGMOD .... Special Interest Group on Management of Data

SGML .... Standardized General Markup Language

SWSS................. Semantic Web Working Symposium

TODS................. Transactions on Database Systems

TOPLAS .... Transactions on Programming Languages and Systems

TU...................... Technische Universität

u. .... und

u.a. .... unter anderem

u.ä. .... und ähnliche(s)

u.U. .... unter Umständen

UML .... Unified Modeling Language

URI .... Uniform Resource Identifier

URL .... Uniform Resource Locator

URN .... Uniform Resource Name

Vgl. / vgl. .... Vergleiche / vergleiche

VLDB .... Very Large Data Bases (Zeitschrift)

W3C .... World Wide Web Consortium

WES................... Workshop on Web Services, e-Business, and the Semantic Web

XML .... Extensible Markup Language

z.B. .... zum Beispiel

Symbolverzeichnis

Im Symbolverzeichnis sind sämtliche Symbole aufgeführt, die in der vorliegenden Arbeit mehrfach verwendet werden. Die Bedeutung von Symbolen, die nur einmal verwendet werden, geht aus dem jeweiligen Argumentationskontext hervor. Darüber hinaus sind im Symbolverzeichnis Ausdrucksmittel ausgeschlossen, die in Beispielen verwendet werden. Daher werden auch die Ausdrucksmittel aus der Fallstudie im Symbolverzeichnis nicht berücksichtigt.

logische Symbole

$|$ .... Trennlinie für Alternativen

$\in$ .... Element von

$\notin$ .... nicht Element von

$\wedge$ .... Konjunktor – logisches „und“

$\veebar$ .... Disjunktor – logisches „entweder ... oder“

$\vee$ .... Adjunktor – logisches „oder“

$\forall$ .... Allquantor („Für alle ...“)

$\underline{\exists}$ .... Einsquantor („Es gibt genau ein ...“)

$\exists$ .... Existenzquantor („Es gibt mindestens ein ...“)

$\rightarrow$ .... objektsprachlicher Subjugatspfeil („Wenn, ... dann ...“)

:- .... PROLOG - Subjugatspfeil

<- .... F-Logic - Subjugatspfeil

$\leftrightarrow$ .... objektsprachlicher Bijugatspfeil („Genau dann, ... wenn ...“)

$\Leftrightarrow$ .... metasprachlicher Bijugatspfeil („Genau dann, ... wenn ...“)

λ .... leeres Wort

Σ .... Summenoperator

$\top$ .... Maximalkonzept

$\perp$ .... Minimalkonzept

$\models$ .... Modellrelation

$\Vdash$ .... Folgerungsrelation

$\vdash$ .... Ableitungsrelation

$\cup$ .... Mengenvereinigungoperator

$\cap$ .... Schnittmengenoperator

$\setminus$ .... Restmengenoperator

$\subseteq$ .... unechte Teilmengenrelation

$\subset$ .... echte Teilmengenrelation

$\equiv$ .... logische Äquivalenzrelation

$\times$ .... kartesisches Produkt

$\sqsubseteq$ .... Subkonzeptrelation

$\curlyvee$ .... Inkompatibilitätsrelation

$\circeq$ .... Äquivalenzrelation

$\preceq$ .... Unterordnungerelation für ontologische Termtupel

$=$ .... gleich

$>$ .... größer

$<$ .... kleiner

$\leq$ .... kleiner oder gleich

$\geq$ .... größer oder gleich

$\#$ .... Kardinalitätsoperator für Multimengen

$\oplus$ .... komponentenweise Addition von Multimengen

θ .... Familie konzeptspezifischer Variablensubstitutionsfunktionen

θ_k .... konzeptspezifische Variablensubstitutionsfunktion

θ_k^x .... bedingte konzeptspezifische Variablensubstitutionsfunktion

2^θ .... Menge aller Variablenssubstitutionen

$[\theta]$ .... Ausdruckssubstitutionsfunktion

Π .... kartesischer Produktoperator

∞ .... unendlich

$\mathbb{R}$ .... Menge der reellen Zahlen

$\mathbb{R}_+$ .... Menge der positiven reellen Zahlen

$\mathbb{R}_-$ .... Menge der negativen reellen Zahlen

$\mathbb{Z}$ .... Menge der ganzen Zahlen

$\mathbb{Z}_+$ .... Menge der positiven ganzen Zahlen

$\mathbb{Z}_-$ .... Menge der negativen ganzen Zahlen

$\mathbb{N}$ .... Menge der natürlichen Zahlen

$\mathbb{N}_+$ .... Menge der positiven natürlichen Zahlen

$\varnothing_M$ .... leere Multimenge

:= .... Zuweisungsoperator

[...] .... Liste

=> .... F-Logic – einwertiger Attributspfeil (Konzeptebene)

=>> .... F-Logic – mengenwertiger Attributspfeil (Konzeptebene)

-> .... F-Logic – einwertiger Attributspfeil (Instanzenebene)

->> .... F-Logic – mengenwertiger Attributspfeil (Instanzenebene)

deskriptive Symbole

$<t_1,...,t_n>$ .... ontologisches Termtupel

$[tr_1...tr_n>$ .... Transitionenfolge

$A(SIG_{KS})$ .... Menge der Strukturen zu einer konventionellen Signatur SIG_{KS}

$A(SIG_{OS})$ .... Menge der Strukturen zu einer ontologischen Signatur SIG_{OS}

$A(SIG_{PL})$ .... Menge der Strukturen zu einer prädikatenlogischen Signatur SIG_{PL}

$A(SIG_{SS})$ .... Menge der Strukturen zu einer sortierten Signatur SIG_{SS}

AKT_{ON} .... Aktivierungsrelation für Transitionen aus Ontologie-Netzen ON

AKT_{ON}^{SF} .... Aktivierungsrelation für höhere Schaltfolgen aus Ontologie-Netzen ON

AKT_{STN} .... Aktivierungsrelation für Transitionen aus S/T-Netzen STN

AKT_{STN}^{SF} .... Aktivierungsrelation für einfache Schaltfolgen aus S/T-Netzen STN

$ALPH_{KS}$ .... konventionelles Alphabet

$ALPH_{KS}*$ .... Menge der Zeichenketten über dem konventionellen Alphabet $ALPH_{KS}$

$ALPH_{META}$ .... metasprachliches Alphabet

$ALPH_{META}*$ ...... Menge der Zeichenketten über dem metasprachlichen Alphabet $ALPH_{META}$

$ALPH_{META}{}^n$ .... Menge der Zeichenketten der Länge n über dem metasprachlichen Alphabet $ALPH_{META}$

$ALPH_{OS}$ .... ontologisches Alphabet

$ALPH_{OS}*$ .... Menge der Zeichenketten über dem ontologischen Alphabet $ALPH_{OS}$

$ALPH_{SS}$ .... sortiertes Alphabet

$ALPH_{SS}*$ .... Menge der Zeichenketten über dem sortierten Alphabet $ALPH_{SS}$

AN_{FR} .... allgemeine Flusskantenannotation

$AN_{FR}^{\bullet}$ .... spezielle Flusskantenannotation

AN_{IR} .... allgemeine Informationskantenannotation

$AN_{IR}^{\bullet}$ .... spezielle Informationskantenannotation

AN_{RS} .... Relationssymbolannotationsfunktion

AN_{ST} .... Stellenannotationsfunktion

AN_{TR} .... Transitionsannotationsfunktion

$ARG_{OPS_{KS}}$ .... Argumenttypisierungsfunktion für Operationssymbole aus konventionellen Signaturen

$ARG_{OPS_{OS}}$ .... Argumenttypisierungsfunktion für Operationssymbole aus ontologischen Signaturen

$ARG_{OPS_{SS}}$ .... Argumenttypisierungsfunktion für Operationssymbole aus sortierten Signaturen

$ARG_{RS_{KS}}$ .... Argumenttypisierungsfunktion für Relationssymbole aus konventionellen Signaturen

$ARG_{RS_{OS}}$ .... Argumenttypisierungsfunktion für Relationssymbole aus ontologischen Signaturen

$ARG_{RS_{SS}}$ .... Argumentfunktion für Relationssymbole aus sortierten Signaturen

$A_{SIG_{KS}}$ .... Struktur zu einer konventionellen Signatur SIG_{KS}

$A_{SIG_{OS}}$ .... Struktur zu einer ontologischen Signatur SIG_{OS}

$A_{SIG_{PL}}$ .... Struktur zu einer prädikatenlogischen Signatur SIG_{PL}

$A_{SIG_{SS}}$ .... Struktur zu einer sortierten Signatur SIG_{SS}

$A_{SIG_{ST}}$ .... Struktur zu einer statischen ontologischen Signatur SIG_{ST}

A_{SIG_z} .... zustandsbezogene Struktur zu einer dynamischen Signatur SIG_{DY}

$ATF_{SIG_{OS}}$ .... Familie konzeptspezifischer Mengen atomarer Terme über einer ontologischen Signatur SIG_{OS}

$ATF_{SIG_{SS}}$ .... Familie sortenspezifischer Mengen atomarer Terme über einer sortierten Signatur SIG_{SS}

$AT_{SIG_{KS}}$ .... Menge der atomaren Terme über einer konventionellen Signatur SIG_{KS}

$AT_{SIG_{OS}}$ .... Menge der atomaren Terme über einer ontologischen Signatur SIG_{OS}

$AT_{SIG_{SS}}$ .... Menge der atomaren Terme über einer sortierten Signatur SIG_{SS}

bel_{KS} .... konventionelle Variablenbelegungsfunktion

$bel_{KS}[x^*/ob_u]$ ..... konventionelle Variablenbelegungsfunktion mit bedingter Belegung der Variablen x*

$belf_{SS}$ .... Familie sortenspezifischer Variablenbelegungsfunktionen

$belf_{SS}[x^*/ob_u])$ ... Familie sortenspezifischer Variablenbelegungsfunktionen mit bedingter Belegung der Variablen x*

bel_s .... sortenspezifische Variablenbelegungsfunktion

$bel_s[x^*/ob_u]$ .... sortenspezifische Variablenbelegungsfunktion mit bedingter Belegung der Variablen x*

$belf_{OS}$ .... Familie konzeptspezifischer Variablenbelegungsfunktionen

$belf_{OS}[x^*/ob_u])$ .. Familie konzeptspezifischer Variablenbelegungsfunktionen mit bedingter Belegung der Variablen x*

bel_k .... konzeptspezifische Variablenbelegungsfunktion

$bel_k[x^*/ob_u]$........ bedingte konzeptsspezifische Variablenbelegungsfunktion mit bedingter Belegung der Variablen x*

bezf .... Familie sprachspezifischer Bezeichnungsfunktionen

bez_{lan} .... sprachspezifische Bezeichnungsfunktion für die Sprache lan

$CF_{SIG_{KS}}$ .... Menge der geschlossenen (konventionellen) Formeln über der konventionellen Signatur SIG_{KS}

$CF_{SIG_{OS}}$ .... Menge der geschlossenen (ontologischen) Formeln über der ontologischen Signatur SIG_{OS}

$CF_{SIG_{SS}}$ .... Menge der geschlossenen (sortierten) Formeln über der sortierten Signatur SIG_{SS}

deff .... Familie sprachspezifischer Definitionsfunktionen

def_{lan} .... sprachspezifische Definitionsfunktion

$ETF_{SIG_{OS}}$ .... Familie konzeptspezifischer Mengen einwertiger Terme über der ontologischen Signatur SIG_{OS}

$EXPR_{SIG_{KS}}$ .... Menge der (konventionellen) Ausdrücke über der konventionellen SIG_{KS}

$EXPR_{SIG_{OS}}$ .... Menge der (ontologischen) Ausdrücke über der ontologischen Signatur SIG_{OS}

$EXPR_{SIG_{SS}}$ .... Menge der (sortierten) Ausdrücke über der sortierten Signatur SIG_{KS}

f .... kontradiktorische Formel

F .... Formel

FB .... Faktenbasis

FB_{EXPL} .... explizite Faktenbasis

FB_{IMPL} .... implizite Faktenbasis

FM .... Formelmenge

FORALL .... algorithmischer Schleifenoperator / Allquantor (F-Logic)

$FORM_{SIG_{DY}}$ .... Menge der Formeln über der (dynamischen) Signatur SIG_{DY}

$FORM_{SIG_{KS}}$ .... Menge der (konventionellen) Formeln
über der konventionellen Signatur SIG_{KS}

$FORM_{SIG_{OS}}$ .... Menge der (ontologischen) Formeln
über der ontologischen Signatur SIG_{OS}

$FORM_{SIG_{PL}}$ .... Menge der (prädikatenlogischen) Formeln
über der prädikatenlogischen Signatur SIG_{PL}

$FORM_{SIG_{SS}}$ .... Menge der (sortierten) Formeln
über der sortierten Signatur SIG_{SS}

$FORM_{SIG_{ST}}$ .... Menge der Formeln über der (statischen) Signatur SIG_{ST}

FR .... Flussrelation

$fvar_{KS}$ .... freie Variablenanteilsfunktion für konventionelle Formeln

$fvar_{OS}$ .... freie Variablenanteilsfunktion für ontologische Formeln

$fvar_{SS}$ .... freie Variablenanteilsfunktion für sortierte Formeln

G .... allgemeine Gewichtungsfunktion

$G^{\bullet}$ .... spezielle Gewichtungsfunktion

$GF_{SIG_{KS}}$ .... Menge der (konventionellen) Grundformeln
über der konventionellen Signatur SIG_{KS}

$GF_{SIG_{OS}}$ .... Menge der (ontologischen) Grundformeln
über der ontologischen SIG_{OS}

$GF_{SIG_{SS}}$ .... Menge der (sortierten) Grundformeln
über der sortierten Signatur SIG_{SS}

$GTF_{SIG_{SS}}$ .... Familie sortenspezifischer Grundtermmengen
über der sortierten Signatur SIG_{SS}

$GTF_{SIG_{OS}}$ .... Familie konzeptspezifischer Grundtermmengen
über der ontologischen Signatur SIG_{OS}

GT_k .... Menge (ontologischer) Grundterme zum Konzept k

GT_s .... Menge (sortierter) Grundterme zur Sorte s

$GT_{SIG_{KS}}$ .... Menge der (konventionellen) Grundterme
über der konventionellen Signatur SIG_{KS}

$GTTF_{SIG_{OS}}$ .... Familie von Mengen (ontologischer) Grundtermtupel
über der ontologischen Signatur SIG_{OS}

$GTT_{SIG_{OS}}$ .... Menge der (ontologischen) Grundtermtupel
über der ontologischen Signatur SIG_{OS}

GTT_w .... Menge der ontologischen Grundtermtupel zur Konzeptfolge w

HOM .... Homonymrelation

IB_{ON} .... Integritätsbedingung für Ontologie-Netze ON

IB_{PN} .... Integritätsbedingung für allgemeine Petri-Netze PN

IB_{STN} .... Integritätsbedingung für Stelle/Transition-Netze STN

If,.then,.else ...... algorithmische Klausel

$IND_{SIG_{KS}}$ .... Menge der Individuensymbole
über der konventionellen Signatur SIG_{KS}

$INDF_{SIG_{SS}}$ .... Familie sortenspezifischer Individuensymbolmengen
über der sortierten SIG_{SS}

IND_s .... sortenspezifische Individuensymbolmenge

$INDF_{SIG_{OS}}$ .... Familie konzeptspezifischer Individuensymbolmengen
über der ontologischen SIG_{OS}

IND_k .... konzeptspezifische Individuensymbolmenge

$INF_{SIG_{OS}}$ .... Menge von Inferenzregeln über der ontologischen Signatur SIG_{OS}

$INT_{SIG_{OS}}$ .... Menge von Integritätsregeln über der ontologischen Signatur SIG_{OS}

IR .... Informationsrelation

I_K .... Interpretationsfunktion für Konzepte

IF_{KS} .... Familie der Interpretationsfunktionen aus einer SIG_{KS}-Struktur

IF_{OS} .... Familie der Interpretationsfunktionen aus einer SIG_{OS}-Struktur

IF_{SS} .... Familie der Interpretationsfunktionen aus einer SIG_{SS}-Struktur

IF_{ST} .... Familie der statischen Interpretationsfunktionen

IF_{TT} .... Familie konzeptfolgenspezifischer Termtupelauswertungsfunktionen

I_{OPS} .... Interpretationsfunktion für Operationssymbole

I_{RS} .... Interpretationsfunktion für Relationssymbole

$I_{RS_{ST}}$ .... Interpretationsfunktion für statische Relationssymbole

I_{RS_z} .... zustandsbezogene Interpretationsfunktion
.... für dynamische Relationssymbole

I_S .... Interpretationsfunktion für Sorten

I_w .... Termtupelauswertungsfunktion zur Konzeptfolge w

ITF_{SS} .... Familie der sortenspezifischen Termauswertungsfunktionen

ITF_{OS} .... Familie der konzeptspezifischen Termauswertungsfunktionen

IT_{KS} .... Termauswertungsfunktion für konventionelle Terme

IT_k .... konzeptspezifische Termauswertungsfunktion für ontologische Terme

IT_s .... sortenspezifische Termauswertungsfunktion für sortierte Terme

K^* .... Menge der Konzeptfolgen

K^+ .... Menge der nicht-leeren Konzeptfolgen

$k,k_1,...,k_n$ .... Konzepte

$(k_1....k_n)$ .... Konzeptfolge

K_{DT} .... Menge der Datenkonzepte

K_{EW} .... Menge der einwertigen Terme

K_{MW} .... Menge der mengenwertigen Terme

k_n .... Knoten

$KODM_{SIG_{OS}}$ .... Menge widersprüchlicher (ontologischer) Formelmengen über SIG_{OS}

$KOD_{SIG_{OS}}$ .... Menge widersprüchlicher (ontologischer) Formeln über SIG_{OS}

$KONF_{SIG_{OS}}$ .... Familie konzeptspezifischer Konstantensymbolmengen über SIG_{OS}

$KONF_{SIG_{SS}}$ .... Familie sortenspezifischer Konstantensymbolmengen über SIG_{SS}

KON_k .... konzeptspezifische Konstantensymbolmenge

KON_s .... sortenspezifische Konstantensymbolmenge

$KON_{SIG_{KS}}$ .... Menge der Konstantensymbole über SIG_{KS}

$KON_{SIG_{SS}}$ .... Menge der Konstantensymbole über SIG_{SS}

$KON_{SIG_{OS}}$ .... Menge der Konstantensymbole über SIG_{OS}

KP .... Kapazitätsfunktion

lan .... natürliche Sprache

len_{ALPH} .... Längenfunktion für metasprachliche Ausdrücke

len_{ON} .... Längenfunktion für (höhere) Schaltfolgen aus Ontologie-Netzen ON

len_{STN} .... Längenfunktion für (einfache) Schaltfolgen aus S/T-Netzen STN

lit .... Literal

$M_z(st_m)$ .... zustandsbezogene Markierung der Stelle st_m

M_0 .... Anfangsmarkierung

MEN .... Mengenfunktion

M_f .... Folgemarkierung

MF_{ON} .... Familie möglicher Markierungen in Ontologie-Netzen ON

MF_{STN} .... Familie möglicher Markierungen S/T-Netzen STN

MOD_{KS} .... Modellfunktion für konventionelle Formeln

MOD_{OS} .... Modellfunktion für ontologische Formeln

MOD_{PL} .... Modellfunktion für prädikatenlogische Formeln

MOD_{SS} .... Modellfunktion für sortierte Formeln

M_r .... Referenzmarkierung

M_r [...> M_f .... Übergang von einer Referenzmarkierung M_r
zu einer Folgemarkierung M_f

$MTF_{SIG_{OS}}$ .... Familie konzeptspezifischer Mengen
mengenwertiger Terme über einer ontologischen SIG_{OS}

mult .... Multimenge

mult(a) .... Multiplizität des Elementes a in der Multimenge mult

MULT(A) .... Menge der Multimengen über der Trägermenge A

M_z .... zustandsbezogene Markierung / Zwischenmarkierung

NB_{FR} .... Nachbereichsfunktion für die Flussrelation FR

NB_{KN} .... Nachbereichsfunktion für Knoten

NB_{ST} .... Nachbereichsfunktion für Stellen

NB_{TR} .... Nachbereichsfunktion für Transitionen

OB .... prädikatenlogisches Universum

OBF_{OS} .... Familie konzeptspezifischer Objektmengen

OBF_{SS} .... Familie sortenspezifischer Objektmengen

OB_k .... Objektmenge zum Konzept k

OB_s .... Objektmenge zur Sorte s

$ob_u, ob_1, ..., ob_U$ .... Individuen

$OF_{SIG_{KS}}$ .... Menge der offenen Formeln
über einer konventionellen Signatur SIG_{KS}

$OF_{SIG_{OS}}$ .... Menge der offenen Formeln
über einer ontologischen Signatur SIG_{OS}

$OF_{SIG_{SS}}$ .... Menge der offenen Formeln
über einer sortierten Signatur SIG_{SS}

$O_i, O_1, ..., O_I$ .... Operationssymbol

$o_i, o_1, ..., o_I$ .... Operation

OPF .... Operationenfamilie

OPS .... Menge der Operationssymbole

PN .... allgemeines Petri-Netz

$pot(A)$ .... Potenzmenge über der Menge A

$pot_{+}(A)$ .... leermengenfreie Potenzmenge über Menge A

$R,R_1,...,R_J$ .... Relationssymbol

$r,r_1,...,r_J$ .... Relation

$REG_{SIG_{PL}}$ .... Regeln über der prädikatenlogischen Signatur SIG_{PL}

RF .... Relationenfamilie

RF_{DY} .... Familie dynamischer Relationen

RF_{ST} .... Familie statischer Relationen

RF_z .... zustandsbezogene Relationenfamilie

RS .... Menge der Relationssymbole

RS_{DY} .... Menge der dynamischen Relationssymbole

$\underline{RS}_{DY}$ .... Menge der negativen dynamischen Relationssymbole

RS_{ST} .... Menge der statischen Relationssymbole

$\underline{RS}_{ST}$ .... Menge der negativen statischen Relationssymbole

S .... Menge der Sorten

S^* .... Menge der Sortenfolgen

$s,s_1,...,s_n$ .... Sorten

S^+ .... Menge der nicht-leeren Sortenfolgen

$s_1...s_n$ .... Sortenfolge

sf .... Schaltfolge

SIG_{KS} .... konventionelle Signatur

SIG_{OS} .... ontologische Signatur

SIG_{PL} .... prädikatenlogische Signatur

SIG_{SS} .... sortierte Signatur

$SPEZ_{OS}$ .... ontologische Spezifikation / Ontologie

$SPEZ_{PL}$ .... prädikatenlogische Spezifikation

SR_{ON} .... Schaltfunktion für Transitionen aus Ontologie-Netzen ON

SR_{ON}^{SF} .... Schaltfunktion für höhere Schaltfolgen aus Ontologie-Netzen ON

SR_{STN} .... Schaltfunktion Transitionen aus Stelle/Transition-Netze STN

SR_{STN}^{SF} .... Schaltfunktion für einfache Schaltfolgen
aus Stelle/Transition-Netzen STN

ST .... Stellenmenge

$st_1,...,st_m$ .... Stellen

STN .... Stelle/Transition-Netz

ST_{NEG} .... Menge der negativen Stellen

ST_{POS} .... Menge der positiven Stellen

SYN .... Synonymrelation

$t,t_1,...,t_n$ .... Terme

$TERMF_{SIG_{OS}}$ ...... Familie der konzeptspezifischen Termmengen über einer ontologischen Signatur SIG_{OS}

$TERMF_{SIG_{SS}}$ ...... Familie der sortenspezifischen Termmengen über einer sortierten Signatur SIG_{SS}

$TERM_k$ .... konzeptspezifische Termmenge

$TERM_s$ .... sortenspezifische Termmenge

$TERM_{SIG_{KS}}$ .... Menge der Terme über einer konventionellen Signatur SIG_{KS}

$TERM_{SIG_{KS}}$ .... Menge der Terme über einer konventionellen Signatur SIG_{KS}

$TERM_{SIG_{OS}}$ .... Menge der Terme über einer ontologischen Signatur SIG_{OS}

$TERM_{SIG_{SS}}$ .... Menge der Terme über einer sortierten Signatur SIG_{SS}

tf_{KS} .... Teilformelfunktion für konventionelle Formeln

tf_{OS} .... Teilformelfunktion für ontologische Formeln

tf_{SS} .... Teilformelfunktion für sortierte Formeln

TM .... Theoremfunktion

TN .... Teilnetz

TR .... Transitionenmenge

TR* .... Menge der Transitionsfolgen

$tr_1,...,tr_n$ .... Transitionen

tr_n^- .... transitionsspezifische Löschfunktion

tr_n^+ .... transitionsspezifische Erzeugungsfunktion

$trans_{INF}$ .... Inferenztransitionszuordnungsfunktion

$trans_{INT}$ .... Integritätstransitionszuordnungsfunktion

TR_{DEKL} .... Menge der deklarativen Transitionen

TR_{INF} .... Menge der Inferenztransitionen

TR_{INT} .... Menge der Integritätstransitionen

TR^n .... Menge der Transitionsfolgen der Länge n

TR_{PROZ} .... Menge der prozeduralen Transitionen

$TT_{SIG_{OS}}$ .... Menge der Termtupel über einer ontologischen Signatur SIG_{OS}

TT_w .... Menge der zur Konzeptfolge w zugeordneten Termtupel

$typ_{OPS_{KS}}$ .... Typisierungsfunktion für Operationssymbole in konventionellen Signaturen

$typ_{OPS_{OS}}$ .... Typisierungsfunktion für Operationssymbole in ontologischen Signaturen

$typ_{OPS_{SS}}$ .... Typisierungsfunktion für Operationssymbole in sortierten Signaturen

$typ_{RS_{DY}}$ .... Typisierungsfunktion für dynamische Relationssymbole

$typ_{RS_{KS}}$ .... Typisierungsfunktion für Relationssymbole in konventionellen Signaturen

$typ_{RS_{OS}}$ .... Typisierungsfunktion für Relationssymbole in ontologischen Signaturen

$typ_{RS_{SS}}$ .... Typisierungsfunktion für Relationssymbole in sortierten Signaturen

$typ_{RS_{ST}}$ .... Typisierungsfunktion für statische Relationssymbole

$typ_{T_{OS}}$ .... Typisierungsfunktion für ontologische Terme

VAR .... Variablenmenge

$var_{F_{KS}}$ .... Variablenanteilsfunktion für konventionelle Formeln

$var_{F_{OS}}$ .... Variablenanteilsfunktion für ontologische Formeln

$var_{F_{SS}}$ .... Variablenanteilsfunktion für sortierte Formeln

var_{IN} .... Eingangsvariablenfunktion für Transitionen

var_{OUT} .... Ausgangsvariablenfunktion für Transitionen

$VARF_{SIG_{OS}}$ .... Familie konzeptspezifischer Variablenmengen über einer ontologischen Signatur SIG_{OS}

$VARF_{SIG_{SS}}$ .... Familie sortenspezifischer Variablenmengen über einer sortierten Signatur SIG_{SS}

$var_{T_{KS}}$ .... Variablenanteilsfunktion für konventionelle Terme

$var_{T_{OS}}$ .... Variablenanteilsfunktion für ontologische Terme

$var_{T_{SS}}$ .... Variablenanteilsfunktion für sortierter Terme

var_{TT} .... Variablenanteilsfunktion für (ontologische) Termtupel

VB_{FR} .... Vorbereichsfunktion für die Flussrelation FR

VB_{KN} .... Vorbereichsfunktion für Knoten

VB_{ST} .... Vorbereichsfunktion für Stellen

VB_{TR} .... Nachbereichsfunktion für Transitionen

$VF_{SIG_{KS}}$ .... Menge der variablen Formeln über einer konventionellen Signatur SIG_{KS}

$VF_{SIG_{OS}}$ .... Menge der variablen Formeln über einer ontologischen Signatur SIG_{OS}

$VF_{SIG_{SS}}$ .... Menge der variablen Formeln über einer sortierten Signatur SIG_{SS}

$VTF_{SIG_{OS}}$ .... Familie konzeptspezifischer Termmengen über einer ontologischen Signatur SIG_{OS}

$VTF_{SIG_{SS}}$ .... Familie sortenspezifischer Termmengen über einer sortierten Signatur SIG_{SS}

$VT_{SIG_{KS}}$ .... Menge der variablen Terme über einer konventionellen Signatur SIG_{KS}

w .... tautologische Formel

WBS .... ontologiegestützte Wissensbasis

$x, x_1, \ldots, x_q$ .... Variablen

$ZIEL_{OPS_{KS}}$ .... Zieltypisierungsfunktion für Operationssymbole in konventionellen Signaturen

$ZIEL_{OPS_{OS}}$ .... Zieltypisierungsfunktion für Operationssymbole in ontologischen Signaturen

$ZIEL_{OPS_{SS}}$ .... Zieltypisierungsfunktion für Operationssymbole in sortierten Signaturen

ZMF_{ON} .... Familie zulässiger Markierungen in einem Ontologie-Netz ON

ZMF_{STN} .... Familie zulässiger Markierungen in einem S/T-Netz STN

$ZTF_{SIG_{OS}}$ .... Familie konzeptspezifischer Mengen zusammengesetzter Terme über der ontologischen Signatur SIG_{OS}

$ZTF_{SIG_{SS}}$ .... Familie sortenspezifischer Mengen zusammengesetzter Terme über der sortierten Signatur SIG_{SS}

$ZT_{SIG_{KS}}$ .... Menge der zusammengesetzten Terme über der konventionellen Signatur SIG_{KS}

$ZT_{SIG_{OS}}$ .... Menge der zusammengesetzten Terme über der ontologischen Signatur SIG_{OS}

Abbildungsverzeichnis

Seite

Tabellenverzeichnis

Seite

1 Exposition

1.1 Wissenschaftliche Problemstellung

Die Bedeutung *kooperativer Informationssysteme*[1] für die Wirtschaftswissenschaften hat in den letzten Jahren stetig zugenommen.[2] Das zunehmende Interesse ist vorrangig auf *informationstechnische* Entwicklungen im Bereich kooperativer Informationssysteme zurückzuführen. Insbesondere durch die Entwicklung der *Extensible Markup Language* (XML)[3] konnte die Interoperabilität von kooperativen Informationssystemen in hohem Maße gesteigert werden.[4] Mit XML-gestützten Sprachen, die auf die intendierten Anwendungsbereiche kooperativer Informationssysteme zugeschnitten sind, kann

---

1) Als *kooperative Informationssysteme* werden in der vorliegenden Arbeit solche Informationssysteme verstanden, die zwecks Erreichung gemeinsamer Ziele arbeitsteilig mit anderen – ebenso kooperativen Informationssystemen – zusammenwirken; vgl. DE MICHELIS ET AL. (1997), S. 3 ff. Beispielsweise gehören hierzu Supply-Chain-Management-Systeme und Web Services. Zentrales Element kooperativer Informationssysteme ist der *Austausch* koordinierungsrelevanten Wissens zwecks Erreichung der gemeinsamen Ziele; vgl. KLUSCH (1996), S. 11 ff.; VERHAREN/DIGNUM (1997), S. 195 f. Die Prozesse des Wissensaustausches werden als *Kommunikation* bezeichnet; vgl. RAPAPORT (2003), S. 402. Mit dieser Arbeitsdefinition für den Begriff kooperative Informationssysteme wird ein weites Spektrum von Informationssystemen umfasst.

2) Die Zunahme der Bedeutung von kooperativen Informationssystemen für die Wirtschaftswissenschaften kann einerseits *unmittelbar* anhand von Veröffentlichungen ausgemacht werden, die einen *direkten* Bezug zu kooperativen Informationssystemen aufweisen; vgl. hierzu die zuvor aufgeführten Quellen zu kooperativen Informationssystemen. Andererseits kann die Bedeutungszunahme *mittelbar* anhand von Veröffentlichungen zu Themen identifiziert werden, wodurch ein *indirekter* Bezug zu kooperativen Informationssystemen hergestellt wird. Hierzu gehören vor allem Ausarbeitungen zu *zwischenbetrieblichen Kooperationen*. Als eine zwischenbetriebliche Kooperation wird jede Form arbeitsteiligen Zusammenwirkens von Unternehmen bezeichnet, bei denen zwar eine rechtliche Unabhängigkeit der beteiligten Unternehmen voneinander besteht, allerdings eine gegenseitige Abhängigkeit bezüglich des gemeinsamen Wirkens vorherrscht; vgl. BALLING (1997), S. 14; FONTANARI (1996), S. 36; VON DER OELSNITZ (2003), S. 186. Ein besonderes Augenmerk wurde in diesem Umfeld auf *Unternehmensnetzwerke* im Allgemeinen und *virtuelle Unternehmen* im Besonderen gerichtet; zu Unternehmensnetzwerken vgl. SYDOW (1992), S. 82 ff.; zu virtuellen Unternehmen vgl. MERTENS ET AL. (1998), S. 3; WÜTHRICH/PHILIPP (1998), S. 204. Als „Enabler" für virtuelle Unternehmen wird oftmals das Leistungspotenzial von Informations- und Kommunikationstechnologien angesprochen; vgl. KLEIN (1996), S. 1 und 38 ff.

3) XML ist eine *Grammatik* zur Konstruktion strukturierter Dokumente; vgl. BERSTEL/BOASSON (2002), S. 650 ff.; BRAY ET AL. (2004), Abschnitt 2. Die Bezeichnung „Language" ist insofern missverständlich, weil XML selbst keine sprachlichen Ausdrucksmittel zur Verfügung stellt, sondern nur Vorgaben trifft, wie Ausdrücke in wohlgeformten Dokumenten zu spezifizieren sind. Anhand dieser Vorgaben werden durch XML – im Gegensatz zu der bisweilen im Internet dominierenden Auszeichnungssprache *Hypertext Markup Lanugage* (HTML) – die *Inhalte* von Dokumenten von ihrer *Darstellung* getrennt.

4) Vgl. ANTONIOU/VAN HARMELEN (2004) S. 23 ff.; KLAPSING (2003), S. 2.

gewährleistet werden, dass die beteiligten Informationssysteme eine gemeinsame *syntaktische* Grundlage für ihre koordinierungsrelevante Kommunikation haben.

Dem steigenden Interesse an kooperativen Informationssystemen einerseits stehen die Entwicklungen im Bereich des *Semantic Web*[1)] andererseits gegenüber. Beim Semantic Web handelt sich um ein Szenario, dessen zentrales Element *Ontologien* sind. Im Gegensatz zu vielen traditionellen Modellierungsmethoden werden im Rahmen der *ontologiegestützten Modellierung* sprachliche Ausdrucksmittel verwendet, deren *Bedeutung* formal präzisiert ist.

Eine solche Präzisierung der Bedeutung sprachlicher Ausdrucksmittel ist besonders dann von Interesse, wenn Akteure mit unterschiedlichen sprachlichen Hintergründen zwecks Koordinierung ihres arbeitsteiligen Zusammenwirkens in Kommunikation miteinander treten. Daher werden Ontologien auch über webbasierte Systeme hinaus in Szenarien untersucht, in denen Akteure mit unterschiedlichen Sprach- und Wissenshintergründen in Kommunikation miteinander treten. Für die Wirtschaftswissenschaften sind hierbei insbesondere die Anwendungspotenziale von Ontologien für die Bereiche der *Unternehmensmodellierung*[2)], des inner- und überbetrieblichen *Wissensmanagements*[3)] und des *E-Commerce*[4)] von Interesse. Über die genannten primär wirtschafts-

---

1) Das *Semantic Web* wird vorrangig durch eine Initiative des *World Wide Web Consortiums* (W3C) entwickelt. In mehreren Projekten werden hierbei Technologien diskutiert, um das derzeitige Internet um eine zusätzliche „bedeutungstragende“ Schicht zu erweitern. Vgl. hierzu ANTONIOU/VAN HARMELEN (2004), S. 7 ff.; CROW/ SHADBOLDT (2001), S. 158 ff.; DING ET AL. (2002), S. 210 ff.; HEFLIN (2001), S. 24 ff.; HESSE (2002), S. 478; KIM (2002), S. 52; KLAPSING (2003), S. 2 ff.; MÄDCHE/STAAB (2001), S. 72 ff.; MOTTA ET AL. (2000), S. 1072 ff.; USCHOLD (2002), S. 87 ff.

2) Als *Unternehmensmodell* wird die Repräsentation der betrieblichen Welt aus einer konzeptionellen Sichtweise verstanden werden; vgl. BERTRAM (1996), S. 83; USCHOLD ET AL. (1998), S. 32. Unternehmensmodelle sind eine auf die Beschreibung von Unternehmen gerichtete Sonderform *konzeptioneller Modelle*; vgl. BOMAN ET AL. (1997), S. 193. Im Bereich der *ontologiegestützten Unternehmensmodellierung* weisen die Arbeiten aus dem *TOVE-Projekt* (Toronto Virtual Enterprise) (vgl. GRUNINGER (1997), S. 370 ff.; GRUNINGER ET AL. (2000), S. 382 ff.) und dem *Enterprise-Projekt* (vgl. USCHOLD ET AL. (1998), S. 33 ff.) den höchsten Reifegrad auf. In beiden Projekten sind Ontologien konstruiert worden, mit denen sowohl aufbau- als auch ablauforganisatorische Aspekte von Unternehmen modelliert werden können. Für weitere Ontologien zu Zwecken der Unternehmensmodellierung vgl. MENZEL (1997), S. 73 ff.

3) Vgl. ABECKER/VAN ELST (2004), S. 439 ff.; DECKER (2002), S. 71 ff.; FENSEL (2001A), S. 19 ff.; MIZOGUCHI/ KITAMURA (2001), S. 19 ff.; MULHOLLAND ET AL. (2001), S. 358 ff.; O´LEARY (1998), S. 37 ff.

4) Vgl. DING ET AL. (2004), S. 595 ff.; FENSEL (2001), S. 9; FENSEL (2001A), S. 47 ff.; IZUMI/YAMAGUCHI (2002), S. 79 ff.; LEICH (2002), S. 103 ff.; OMELAYENKO (2002), S. 264 ff.

wissenschaftlichen Domänen hinaus werden Ontologien auch in anderen Disziplinen thematisiert.[1)]

Aufgrund der hohen Bedeutung sowohl von kooperativen Informationssystemen als auch von Ontologien überrascht es nicht, dass die ontologiegestützte Modellierung kooperativer Informationssysteme vermehrt diskutiert wird.[2)] Im Gegensatz zu XML-gestützten Sprachen, durch die lediglich die *syntaktische* Interoperabilität von kooperativen Informationssystemen gewährleistet werden kann, lässt sich mit Hilfe von Ontologien auch eine *semantische* Interoperabilität sicherstellen.[3)] Die semantische Interoperabilität ist notwendig, um Sprachdivergenzen bei der koordinierungsrelevanten Kommunikation der beteiligten Informationssysteme entweder gar nicht erst aufkommen zu lassen oder im Nachhinein zu beheben.

Im Rahmen der koordinierungsrelevanten Kommunikation zwischen kooperativen Informationssystemen ist oftmals neben dem Austausch von Wissen bezüglich *statischer* Phänomene auch der Austausch von Wissen bezüglich *dynamischer* Phänomene notwendig.[4)] Das ist genau dann der Fall, wenn das koordinierungsrelevante Wissen der Akteure sich nicht nur über statische, sondern auch über dynamische Phänomene erstreckt.[5)] Um sowohl das Wissen bezüglich statischer als auch das Wissen bezüglich

---

1) Vgl. HAHN/SCHULZ (2004), S. 133 ff. (Medizin); MCENTIRE (2002), S. 77 ff. (Biologie); STEVENS ET AL. (2004), S. 639 ff. (Biologie); VAN HEIJST ET AL. (1995), S. 235 ff. (Medizin); JOUVE ET AL. (2003), S. 349 ff.; VISSER/ BENCH-CAPON (1998) (Jurispudenz).

2) Vgl. WROE ET AL. (2003), S. 200 ff. (in Bezug auf *Web Services*). Zur Bedeutung von Ontologien für die Modellierung im Allgemeinen vgl. GUARINO (1995), S. 631 ff.; GUIZZARDI ET AL. (2002), S. 65 ff.; MYLOPOULOS ET AL. (1997A), S. 293; PFEIFER (2000), S. 231; SUGUMARAN/STOREY (2002), S. 252 ff.; WIMMER/ WIMMER (1992), S. 387 ff. Zur Modellierung kooperativer Informationssysteme im Allgemeinen vgl. PREECE ET AL. (2001), S. 175 ff.; SCANNAPIECO ET AL. (2004), S. 553 ff.

3) Syntaktische und semantische Interoperabilität können als „Gegenpole" zu syntaktischer bzw. semantischer *Heterogenität* aufgefasst werden; vgl. VISSER ET AL. (1999), S. 668 ff. Syntaktische Heterogenität zwischen Wissensbeständen liegt dann vor, wenn zur Wissensrepräsentation Sprachen mit unterschiedlichen Grammatiken verwendet werden. Eine semantische Heterogenität zwischen Wissensbeständen liegt hingegen dann vor, wenn die Wissensbestände auf unterschiedlichen Konzeptualisierungen basieren. Eine divergierende Konzeptualisierung kann sich einerseits in ambigen Bezeichnungen (Homonyme und Synonyme) und andererseits in strukturellen Konflikten äußern. Beispielsweise liegen divergierende Konzeptualisierungen vor, wenn der gleiche Begriff in zwei Wissensbeständen auf gegenseitig ausschließende Weise *taxonomisch* eingeordnet ist.

4) Vgl. DE MICHELIS ET AL. (1997), S. 23 ff.

5) Beispielsweise wird in der Regel an Modelle von *Web Services* – als eine Form kooperativer Informationssysteme – die Anforderung gestellt, auch *Transaktionen* repräsentieren zu können, die mit dem jeweiligen Dienst vorgenommen werden können und durch die eine dynamische Veränderung von Informationssystemzuständen ermöglicht wird; vgl. MARTIN ET AL. (2004), Abschnitt 1; PAPAZOGLOU (2002), S. 155 ff.; PAPAZOGLOU (2003), S. 50 ff. Derzeit wird hierfür vorrangig die XML-gestützte Sprache *Business Process Execution Language for Web Services* (BPEL) verwendet; vgl. PAPAZOGLOU (2003), S. 67 ff.

dynamischer Phänomene repräsentieren zu können, muss ein Konzept zur Modellierung kooperativer Informationssysteme Ausdrucksmittel für beide Wissensarten zur Verfügung stellen. Es müssen einerseits Ausdrucksmittel zur Verfügung stehen, mit denen Zustände eines Modells ausgedrückt werden können. Andererseits müssen Ausdrucksmittel zur Verfügung stehen, um Variationen der Modellzustände auszudrücken.

Das *wissenschaftliche Problem* der vorliegenden Arbeit knüpft an diese Anforderung an Konzepte zur Modellierung kooperativer Informationssysteme an. Um das koordinierungsrelevante Wissen für kooperative Informationssysteme repräsentieren zu können, muss ein Modellierungskonzept eine *integrative* Modellierung unterstützten, die *sowohl* statische *als auch* dynamische Phänomene zu repräsentieren in der Lage ist. Die Anwendungsbereiche von Ontologien umfassen jedoch in der Regel rein statische Aspekte, da sie nur für die Repräsentation *deklarativen* Wissens verwendet werden können. Um auch *prozedurales* Wissen repräsentieren zu können, werden Ausdrucksmittel benötigt, mit denen sich Zustandsübergänge in ontologiegestützten Modellen aufgezeigen lassen. Die Repräsentation von Zustandübergängen ist eine wesentliche Komponente der Modellierung von dynamischen Aspekten. Eine solche Repräsentation kann mit Hilfe von Ontologien nur bedingt ausgedrückt werden.[1)] Insofern weisen Ontologien für Zwecke der Modellierung kooperativer Informationssysteme ein *Defizit* auf.[2)]

Die wenigen Arbeiten, die einen Bezug von Ontologien zur Repräsentation dynamischer Phänomene aufweisen, konnten bislang nicht zufrieden stellen. Zumeist sind die Arbeiten auf die Konstruktion von Ontologien beschränkt, die zu Zwecken der Repräsentation von Wissen über dynamische Phänomene verwendet werden können.[3)] Allerdings

---

1) Ontologiegestützte Softwaresysteme können aus diesem Blickwinkel auch als *passive* Systeme bezeichnet werden, da zu ihrer Modifikation *konzeptexogene* Eingriffe notwendig sind. Aus diesem Blickwinkel existiert eine Korrespondenz zwischen konventionellen Datenmodellierungssprachen und Ontologien. Für konventionelle Datenmodellierungssprachen liegen in Form von *aktiven Datenbanken* bereits Ausarbeitungen vor, um eine integrative Wissensrepräsentation zu ermöglichen; vgl. DITTRICH/GATZIU (2000), S. 15 ff. Darüber hinaus verfügen auch Systeme auf der Basis von *Produktionsregeln* über eine aktive Komponente; vgl. BEIERLE/KERN-ISBERNER (2000), S. 68 ff.; SCHMID/KINDSMÜLLER (1996), S. 174 ff. Im Gegensatz zu den Event-Condition-Action-Regeln aktiver Datenbanken verfügen Produktionsregeln über keine Komponente in ihrem Rumpf, die das Erfülltsein der Regel von einem systeminternen oder systemexternen *Ereignis* abhängig macht; vgl. LAUSEN ET AL. (1998), S. 74.

2) Dieses Defizit von Ontologien wird auch konventionellen Datenmodellierungssprachen angelastet; vgl. SPECK (2001), S. 1. Aus diesem Grund wurde bereits eine Vielzahl an Modellierungskonzepten entwickelt, in denen bestehende Datenmodellierungssprachen um dynamische Aspekte erweitert werden. Weiter unten werden einige dieser Modellierungskonzepte beispielhaft aufgeführt. Darüber hinaus wird auch für Sprachen zur Konstruktion *Wissensbasierter Systeme* eingefordert, neben statischen Aspekten („non-functional“) auch dynamische Aspekte („functional“) berücksichtigen zu können; vgl. VAN ECK ET AL. (1998), S. 38 ff.

3) vgl. AITKEN/CURTIS (2002), S. 110 f.; BORST/AKKERMANS (1997), S. 373 ff.; GRUENINGER (2004), S. 576 ff.; MARTIN ET AL. (2004), Abschnitt 3; NARAYANAN/MCILRAITH (2002), S. 81 ff.; SEIBT (2001), S. 334 ff.; USCHOLD ET AL. (1998), S. 46 ff.

wird hierdurch lediglich aufgezeigt, wie Begrifflichkeiten zur Repräsentation prozessbezogenen Wissens mit Ontologien formal spezifiziert werden können. Darüber hinaus wird in den Arbeiten allerdings nicht aufgezeigt, wie *Übergänge* zwischen den Zuständen eines ontologiegestützten Modells formal ausgedrückt werden können. Die Variation des Zustands eines ontologiegestützten Modells ist allenfalls durch einen *konzeptexogenen* Eingriff in das Modell möglich. Um Ontologien in ein ganzheitliches Modellierungkonzept einbinden zu können, durch das auch *konzeptendogene* Zustandvariationen in Modellen im Zeitablauf berücksichtigt werden können, bedarf es daher ihrer Erweiterung um eine *prozedurale* Komponente.[1)]

Zur Lösung des wissenschaftlichen Problems wird in der vorliegenden Arbeit als *erstes intendiertes Ergebnis* die Spezifikation einer *Ontologie-Sprache* festgelegt. Um Ontologien in ein integratives Modellierungskonzept einbinden zu können, ist es zunächst notwendig, präzise festzulegen, welche notwendigen und hinreichenden Komponenten Ontologien aufzuweisen haben. Da in einer Ontologie sprachliche Ausdrucksmittel spezifiziert werden, erfolgt eine derartige Festlegung mit *metasprachlichen* Ausdrücken. Welche sprachlichen Ausdrucksmittel genau für eine Ontologie zugelassen werden, wird durch die Ontologie-Sprache festgelegt. Die Ontologie-Sprache umfasst einerseits das metasprachliche *Alphabet* und andererseits *Konstruktionsregeln*, die für die Konstruktion von Ontologien notwendig sind.

Nach der Präzisierung des Verständnisses von Ontologien wird als *zweites intendiertes Ergebnis* die Entwicklung eines *integrativen Modellierungskonzepts* angestrebt. Der Begriff integratives Modellierungskonzept umfasst die Gesamtheit aller Konstruktionsvorschriften, mit denen ontologiegestützte Modelle kooperativer Informationssysteme konstruiert werden können, die sowohl einzelne Zustände der modellierten Informationssysteme als auch prozessbedingte Variationen dieser Zustände umfassen. Zu diesem Zweck wird das formale Gerüst für *Ontologie-Netze* vorgestellt. Bei Ontologie-Netzen handelt es sich um eine Klasse *höherer Petri-Netze*, die mit den Ausdrucksmitteln aus einer Ontologie spezifiziert werden. Hierdurch unterstützen Ontologie-Netze sowohl die Repräsentation von statischen als auch die Repräsentation von dynamischen Phänomenen.

Der Integrationscharakter des Modellierungskonzepts erstreckt sich nicht „nur" auf die Integration von Ontologien und Petri-Netzen durch die Beschriftung von Petri-Netzen mit Ausdrucksmitteln aus einer Ontologie. Darüber hinaus werden Ontologien derart in das integrative Modellierungskonzept eingebettet, dass die ontologiegestützte Modellierung neben einer *deklarativen Semantik* auch um eine *operationale Semantik* erweitert wird. Die deklarative Semantik von ontologiegestützten Modellen wird durch die Menge aller Fakten bestimmt, die sich aus dem Modell als Schlussfolgerungen ableiten lassen. Unbeachtet bleibt hierbei, *wie* die Schlussfolgerungen durchgeführt werden. Die

---

1) Durch die Erweiterung um eine prozedurale Komponente könnten ontologiegestützte Softwaresysteme zu *aktiven* System ausgebaut werden. Zu aktiven Systemen lassen sich in erster Linie *aktive Datenbanken* zählen.

operationale Semantik legt dagegen die zulässigen Operationen fest, mit deren Hilfe Schlussfolgerungen durchgeführt werden können.[1] Sie ist dadurch begründet, dass in Ontologie-Netzen die Variation der Zustände von ontologiegestützten Modellen durch *zustandsändernde Grundoperationen*[2] durchgeführt werden können.

Um sein Leistungspotenzial aufzeigen zu können, ist das *dritte intendierte Ergebnis* der vorliegenden Arbeit die betriebswirtschaftliche *Evaluation des integrativen Modellierungskonzepts*. Der wissenschaftlichen Problemstellung der Arbeit folgend, wird im Rahmen dieser Evaluation zwischen einerseits der Modellierung statischer und andererseits der Modellierung dynamischer Phänomene unterschieden. Zur Evaluation der statischen Modellierungsfähigkeit wird die entwickelte Ontologie-Sprache, die dem ersten intendierten Ergebnis entspricht, anhand eines ersten Anforderungsteilkatalogs mit alternativen Ontologie-Sprachen verglichen. Bezüglich der dynamischen Modellierungsfähigkeit wird die Ausdrucksmächtigkeit des integrativen Modellierungskonzepts anhand eines zweiten Anforderungsteilkatalogs beurteilt.

1.2 Wirtschaftswissenschaftliche Relevanz der Untersuchung

Die wirtschaftswissenschaftliche Relevanz der untersuchten Problemstellung ergibt sich unmittelbar aus den intendierten *Anwendungsbereichen* für das integrative Modellierungskonzept. Hierzu gehört beispielsweise die Konstruktion von *Workflow-Management-Systemen*. Sie können als eine Sonderform kooperativer Informationssysteme betrachtet werden, die auf die Unterstützung des Dokumentflusses ausgerichtet sind.

Die Implementierung eines Workflow-Management-Systems setzt oftmals die Spezifikation der zu unterstützenden Geschäftsprozesse in einem Modell voraus. Um die sowohl syntaktische als auch semantische Interoperabilität von Informationssystemen gewährleisten zu können, die im Rahmen eines Workflow-Management-Systems kooperieren, können Ontologie-Netze von hohem Nutzen sein. Beispielsweise werden im Office-Bereich aktuell vermehrt „Push-Technologien“ diskutiert, durch die Anwender innerhalb ihrer Geschäftsprozesse mit möglicherweise relevanten Informationen versorgt werden. Ontologiegestützte „Push“-Technologien bieten sich hierbei besonders an, um die Benutzer keiner ungefilterten „Informationsflut“ auszusetzen.

Ein weiterer Anwendungsbereich existiert im Rahmen des inner- und überbetrieblichen *Wissensmanagements*. Oftmals scheitert der Austausch koordinierungsrelevanten Wissens an Sprachbarrieren zwischen den beteiligten Akteuren. Auch in diesem Kontext werden vermehrt Ontologien diskutiert, um den Wissensaustausch zu unterstützen.[3] Al-

---

1) Zur operationalen Semantik formalsprachlicher Konstrukte vgl. BEST (1995), S. 54 ff.; EIRUND ET AL. (2000), S. 41 f.

2) Der Begriff der *Grundoperation* wird in Abschnitt 2.1.3.2.3 präzisiert.

3) Vgl. hierzu die eingangs aufgeführten Quellen zur Unterstützung des Wissensmanagements durch Ontologien.

lerdings sind Ontologien in ihrer entkoppelten Verwendung lediglich dazu geeignet, *deklaratives* Wissen zu vermitteln. *Prozedurale* Wissensfragmente entziehen sich hingegen der ontologiegestützten Repräsentation. Im Gegensatz hierzu sind Petri-Netze dazu geeignet, prozedurales Wissen zu repräsentieren.[1)] Dadurch, dass im integrativen Modellierungskonzept sowohl Ontologien als auch Petri-Netze Verwendung finden, können beide Wissensarten adäquat berücksichtigt werden. Das inner- und überbetriebliche Wissensmanagement kann daher mit Hilfe von Ontologie-Netzen auch prozessorientiert ausgerichtet werden. Beispielsweise kann der Zugriff auf Wissensressourcen, der für die Durchführung einer Aktivität notwendig ist, in einem Modell repräsentiert werden.

Von Ontologie-Netzen kann zudem ein hoher Nutzen im Kontext von Multi-Agenten-Systemen (MAS) erwartet werden. Einerseits sind Ontologien dazu geeignet, die Agenten aus einem MAS mit einer gemeinsamen Sprachbasis zu versehen.[2)] Eine solche gemeinsame Sprachbasis ist notwendig, um Sprachbarrieren bei der koordinierungsrelevanten Kommunikation zwischen Agenten zu überwinden.[3)] Andererseits gelten Petri-Netze als geeignete Instrumente zur formalen Spezifikation der Prozesse in MAS.[4)] Ein besonderes Interesse liegt hierbei in der Spezifikation von Prozessen für Agenten, die über betriebliche Grenzen hinaus agieren. Sowohl für traditionelle Kooperationsformen als auch für moderne Supply-Chain-Management-Konzeptionen und virtuelle Unternehmen werden MAS diskutiert.[5)] Der Bedarf nach einer „agentengerechten“ Auszeichnung von Informationen einerseits und nach einer formalen Spezifikation von Prozessen andererseits lässt sich miteinander vereinbaren, wenn Ontologie-Netze herangezogen werden.

Viertens wird durch die Entwicklung des integrativen Modellierungskonzepts die Anschlussfähigkeit der ontologiegestützten Modellierung an vorherrschende Modellierungssprachen bewahrt. Die meisten der aktuell diskutierten Modellierungssprachen sind mittlerweile derart in ganzheitliche Modellierungskonzepte eingebunden worden, dass mit ihrer Hilfe sowohl statische als auch dynamische Phänomene repräsentiert werden können.[6)] Beispielsweise verfügt die Unified Modeling Language (UML) über unterschiedliche Diagrammarten, mit deren Hilfe eine ganzheitliche Modellierung mög-

---

1) Vgl. RÖBBECKE (1995), S. 57 ff.

2) Vgl. HE/LEUNG (2002), S. 272 f.; STEELS (1998), S. 170 ff.

3) Dabei lassen sich grundsätzlich zwei Strategien unterscheiden. Bei der ersten Strategie wird für alle beteiligten Agenten des MAS *eine* Ontologie vorausgesetzt. Dieser Ex-ante-Harmonisierung der Sprachwelten steht eine Ex-post-Harmonisierung gegenüber. Bei dieser zweiten Strategie werden Ausdrücke aus den unterschiedlichen agentenspezifischen Ontologien mittels Übersetzungsverfahren aufeinander abgebildet.

4) Vgl. MUSCHOLL (2001), S. 15 ff.; XU ET AL. (2002), S. 194 ff.

5) Vgl. OUZOUNIS (2001), S. 81 ff.

6) Vgl. BARROS/TER HOFSTEDE (1998), S. 316 ff.

lich ist.[1] Darüber hinaus ist die Petri-Netz-Theorie bereits öfters für die Spezifikation prozessbedingter Operationen auf Datenbanken, die auf relationalen Schemata[2], NR/T-[3], SGML-[4] oder XML-Schemata[5] basieren, herangezogen worden. Auch dies ist ein Indiz dafür, dass das Petri-Netz-Konzept für die Integration von Ontologien geeignet ist. Schliesslich werden Ontologien des Öfteren auch als Unterstützung für die „semantisch zulässige" Konstruktion für einige der o.a. Schemata diskutiert.[6]

1.3 Gang der Untersuchung

Die vorliegende Arbeit ist wie folgt aufgebaut: Im Abschnitt 2 wird das Rahmenwerk für das integrative Modellierungskonzept vorgestellt. Dafür werden zum einen der *theoretische* und zum anderen der *formale* Rahmen entfaltet (Abschnitt 2.1 bzw. 2.2). Während im theoretischen Rahmen die konzeptionellen Basisentscheidungen für die weitere Argumentation vorgestellt werden, umfasst der formale Rahmen die formalsprachlichen Instrumente, die für das Modellierungskonzept benötigt werden. Als theoretischer Rahmen der Arbeit werden ein wissenschaftstheoretischer (Abschnitt 2.1.1), ein systemtheoretischer (Abschnitt 2.1.2) und ein modelltheoretischer Rahmen (Abschnitt 2.1.3) vorgestellt. Innerhalb des modelltheoretischen Rahmens wird ein *Anforderungskatalog* für das Modellierungskonzept aufgebaut (Abschnitt 2.1.3.2.2).

Die Hauptteile der Arbeit bilden die Abschnitte 3 und 4. In Abschnitt 3 werden die einzelnen Bausteine für Ontologie-Netze untersucht. Es handelt sich hierbei einerseits um Ontologien (Abschnitt 3.1.) und andererseits um Petri-Netze (Abschnitt 3.2.). Die beiden Bausteine werden in Abschnitt 4 zum integrativen Modellierungskonzept zusammengeführt. In Abschnitt 5 werden Ontologie-Netze einer Evaluation unterzogen. Hierfür wird der Anforderungskatalog verwendet, der zuvor in Abschnitt 2.1.3.2 erarbeitet wurde. Die Arbeit wird mit einer kritischen Würdigung in Abschnitt 6 abgeschlossen.

Der Gang der Untersuchung ist in der Abbildung 1 grob als Kanal/Instanzen-Netz[7] dargestellt. Die runden Symbole in der Darstellung repräsentieren jeweils die obersten Abschnitte der vorliegenden Arbeit. Durch die rechteckigen Symbole wird der Zusammenhang der Abschnitte für die Argumentation – entsprechend den gerichteten Pfeilen – verdeutlicht.

---

1) Vgl. FOWLER (2004), S. 45 f.

2) Vgl. REISIG (1991), S. 132. ff.

3) Vgl. OBERWEIS (1996), S. 98 ff.

4) Vgl. WEITZ (2000), S. 113 ff.

5) Vgl. LENZ (2003), S. 170 ff.

6) Vgl. STOREY ET AL. (1998), S. 32 ff.; SUGUMARAN/STOREY (2002), S. 254 ff.; ULLRICH ET AL. (2000), S. 94 ff. Zum Verhältnis von Ontologien zu Datenbankschemata vgl. BENCH-CAPON ET AL. (2002), S. 703 ff.; GRUBER (1993), S. 202; OUZZANI ET AL. (2000), S. 368.

7) Zu Kanal/Instanzen-Netzen vgl. OBERWEIS (1996), S. 100.

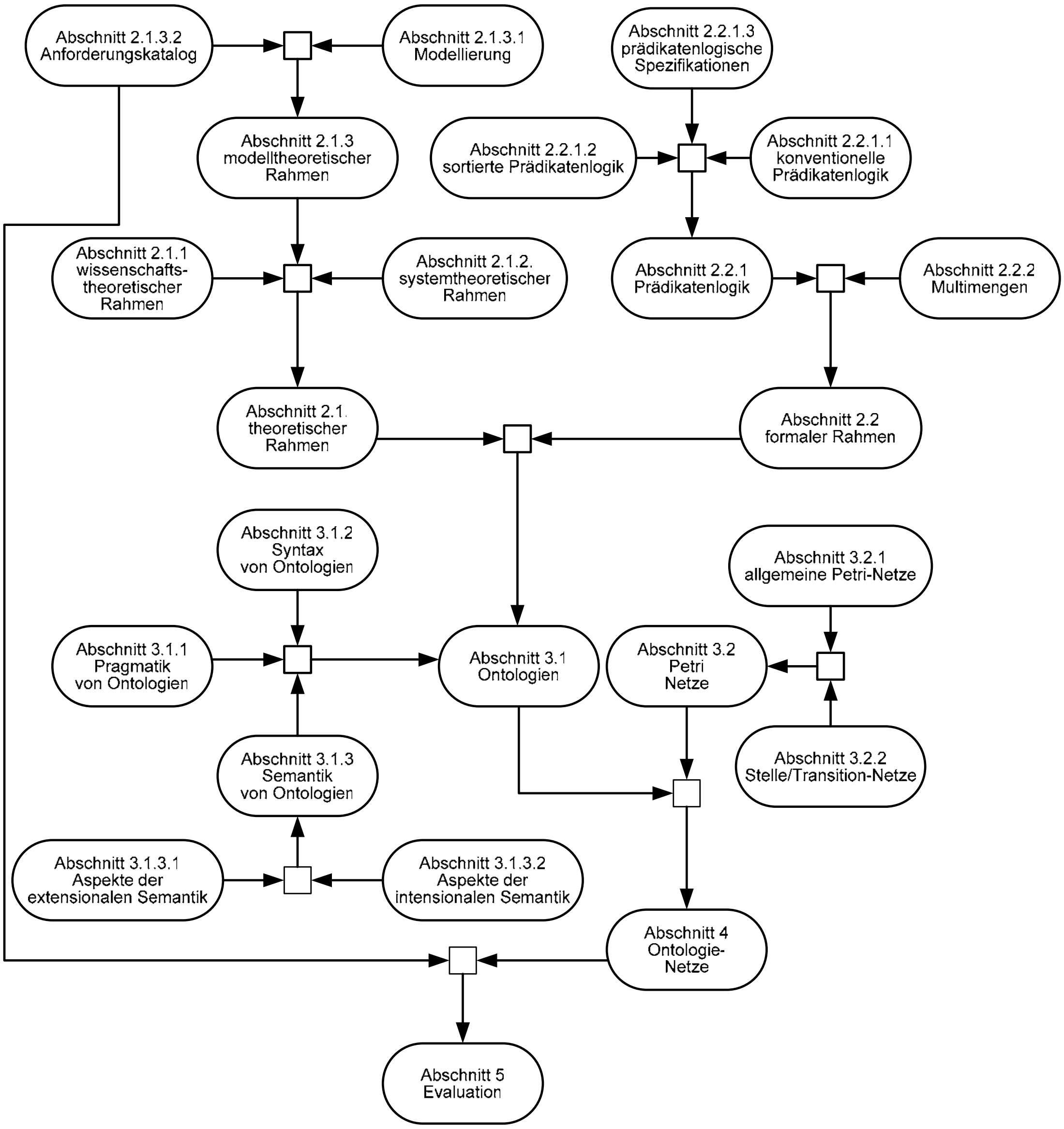

Abbildung 1: Gang der Untersuchung

2 Rahmenwerk für das integrative Modellierungskonzept

2.1 Theoretischer Rahmen des integrativen Modellierungskonzepts

2.1.1 Wissenschaftstheoretischer Rahmen

Der wissenschaftstheoretische Rahmen, innerhalb dessen die vorliegende Arbeit eingeordnet werden kann, erstreckt sich über zwei Aspekte, die sich gegenseitig ergänzen. Der erste Aspekt ist auf den Begriff der Ontologie aus der *Philosophie* bezogen. Die Bezeichnung „Ontologie“ entstammt ursprünglich Disziplinen, deren Wurzeln bis hin zur griechischen Philosophie reichen. Das moderne Verständnis von Ontologien ist hingegen durch eine homonyme Verwendung der Bezeichnung gekennzeichnet.

Aus der Klärung des Begriffs Ontologie ergibt sich insofern ein „positiver Nebeneffekt“, als sich das philosophische Ontologie-Verständnis für den zweiten Aspekt, der auf die *wissenschaftstheoretische Positionierung* bezogen ist, als hilfreich erweist, um alternative wissenschaftstheoretische Basispositionen abzugrenzen. Aufbauend auf der Begriffsklärung kann der wissenschaftstheoretische Rahmen der vorliegenden Arbeit bestimmt werden. Diese Bestimmung ist insofern von höchster Bedeutung, als ein Anspruch von Ontologiekonstrukteuren auf Realitätsbezogenheit ihrer Ontologien nur vor diesem Hintergrund beleuchtet werden kann. Um diesen Anspruch zu überprüfen, muss die wissenschaftstheoretische Basisposition beurteilt werden, die bei der Konstruktion der Ontologie vorausgesetzt wird.[1)]

Der Begriff *Ontologie* entstammt ursprünglich der Philosophie und entspricht dort dem allgemeinen Teil der *Metaphysik*.[2)] Die Ontologie bildet den Teilbereich der Metaphysik, der sich mit dem *Seienden* und dem *Sein* an sich beschäftigt. Insofern wird im Rahmen der Ontologie durch die Gesamtheit des Seins zu der *Realität* Bezug genom-

---

1) Die wissenschaftstheoretische Positionierung hat eine besondere Bedeutung für die *Evaluation* von Ontologien. Für die Gütebeurteilung einer Ontologie muss nämlich auch die wissenschaftstheoretische Basisposition berücksichtigt werden, die bei der Konstruktion der Ontologie eingenommen wurde. Diese Feststellung kann analog von der Evaluierbarkeit *konzeptioneller Modelle* in Bezug auf die zu Grunde gelegte wissenschaftstheoretische Basisposition übertragen werden. Sie wurde vornehmlich im Rahmen der *Grundsätze ordnungsmäßiger Referenzmodellierung* von SCHÜTTE vertreten; vgl. SCHÜTTE (1998), S. 11 ff. Beispielsweise muss analog zu der Evaluation konzeptioneller Modelle für die Evaluation einer Ontologie zunächst geklärt werden, ob ein subjektunabhängiger Zugang zu der Realität für möglich gehalten wird. Wenn ein solcher Zugang entsprechend der eingenommenen Basisposition verwehrt bleibt, kann an eine Ontologie auch nicht der Anspruch gestellt werden, die Realität abzubilden. Obwohl sich der Evaluation von Ontologien mittlerweile eine Vielzahl von Arbeiten gewidmet hat (vgl. GRUBER (1992A), S. 2 ff.; FOX/GRUENINGER (1997), S. 195 ff.; GÓMEZ-PÉREZ (2004), S. 252 ff.; GÓMEZ-PÉREZ (2001), S. 393 ff.), wird eine wissenschaftstheoretische Auseinandersetzung mit Ontologien in diesem Kontext vermisst.

2) Für einen Überblick über die historische Entwicklung der philosophischen Ontologie vgl. SOWA (1995), S. 673 ff.; WEISSMAHR (1991), S. 9 ff.

men. Der Ontologie kann die *Erkenntnistheorie* oder *Epistemologie* gegenübergestellt werden. Im Gegensatz zur Ontologie beschäftigt sich die Erkenntnistheorie nicht mit der (ontischen) *Beschaffenheit* der Realität, sondern mit dem (epistemischen) *Zugang* zu dieser Realität.

Die unterschiedlichen Untersuchungsgegenstände der Ontologie und der Erkenntnistheorie können für eine Differenzierung wissenschaftstheoretischer Basispositionen herangezogen werden. Mit jeder wissenschaftstheoretischen Positionierung sind erstens Annahmen über die Beschaffenheit der Realität (ontologische Perspektive) und zweitens Annahmen über den Zugang zu dieser Realität (epistemologische Perspektive) verbunden. Grundsätzlich können die wissenschaftstheoretischen Basispositionen, die für die Thematik der vorliegenden Arbeit von Relevanz sind, in einerseits *realistische* und andererseits *idealistische* Konzeptionen unterteilt werden.[1] Tabelle 1 gibt einen Überblick über einige wissenschaftstheoretische Basispositionen und deren ontologischen bzw. epistemologischen Perspektiven.

		epistemologische Perspektive	
		idealistisch	realistisch
ontologische Perspektive	idealistisch	radikaler Konstruktivismus	⊘
	realistisch	kritischer Realismus/ gemäßigter Konstruktivismus	naiver Realismus

Tabelle 1: Klassifikation wissenschaftstheoretischer Basispositionen

Einem *ontologischen Realismus* zufolge kann beispielsweise von einer *Realität* ausgegangen werden, die unabhängig von erkennenden Subjekten existiert. Wird diese Sichtweise mit einem *epistemologischen Realismus* kombiniert, so liegt die Sichtweise des *naiven Realismus* vor. Der naive Realismus unterstellt sowohl eine subjektunabhängig existierende Realität als auch die Möglichkeit der ebenso subjektunabhängigen Erkennbarkeit dieser Realität.

1) Zu Überblicken über realistische und idealistische Konzeptionen vgl. SCHÜTTE (1999), S. 219 ff.; WESTERMANN (2000), S. 29 ff.

Wird ein ontologischer Realismus hingegen mit einem *epistemologischen Idealismus*[1)] kombiniert, wird zwar von der subjektunabhängig existierenden Realität, aber nicht von der Möglichkeit ihrer subjektunabhängigen Erkennbarkeit ausgegangen. Sämtliche Spielarten des epistemologischen Idealismus haben hierdurch gemeinsam, dass durch sie die Konstruktionsleistung des erkennenden Subjekts hervorgehoben wird. Dem Ursprung jeder Erkenntnis dem epistemologischen Idealismus entspricht der *Geist* des erkennenden Subjekts. Aussagen über die Realität sind daher aus diesem Blickwinkel immer subjektbezogen zu beurteilen.

Beispielsweise wird die Kombination aus ontologischem Realismus und epistemologischen Idealismus von Anhängern des *kritischen Realismus* vertreten. Zwar gehen auch sie von einer Realität aus, deren Strukturen vorgegeben sind,[2)] allerdings wird der Subjektbezug jeder Erfahrung hervorgehoben. Die Bedeutung des Subjektbezugs wird wiederum dadurch hervorgehoben, dass die Erfahrungen von Subjekten hinsichtlich objektiv gültiger Aussagen über die Realität bewertet werden können. Durch die Bewertbarkeit der Aussagen wird der epistemologischen Sichtweise Rechnung getragen. In dieser Sichtweise wird durch den kritischen Realismus der Grundstein für die Methodologie des *kritischen Rationalismus* – insbesondere POPPERscher Prägung – gelegt, die in den Wirtschaftswissenschaften die weiteste Verbreitung genießt.[3)] Über den kritischen Realismus hinaus wird die Auffassung einer subjektiven Erkenntnis der objektiv vorliegenden Realität auch vom *hypothetischen Realismus* vertreten.

Im Gegensatz zum ontologischen Realismus negiert der *ontologische Idealismus* die Existenz einer subjektunabhängig existierenden Realität. Er ist insbesondere als *radikaler Konstruktivismus* bekannt geworden.[4)] Der radikale Konstruktivismus verneint die subjektunabhängige Realität an sich.[5)] Insofern ist der Zugang zu einer objektiven Realität durch ein Subjekt entsprechend dieser Sichtweise auch nicht möglich. Jedoch wird der radikale Konstruktivismus dadurch, dass seine Sichtweisen mit neurophysiologische Vorgängen – also subjektunabhängig existierenden Realitätssauschnitten – erklärt werden, oftmals einem *Zirkularitätsproblem* ausgesetzt.

Von den vier potenziellen Kombinationsmöglichkeiten von ontologischer und erkenntnistheoretischer Sichtweise einerseits sowie realistischer und idealistischer Positionierung andererseits kann die Kombination aus ontologischem Idealismus und epistemologischen Realismus ausgeschlossen werden. Denn der Ausschluss einer subjektunabhän-

1) Zu einem Überblick über verschiedene Ausprägungen des epistemologischen Idealismus vgl. SCHÜTTE (1999), S. 222 ff.

2) Vgl. POPPER (1994), S. 392 f.

3) Vgl. ALBERT (2000); POPPER (1994).

4) Vgl. GLASERFELD (1987); MATURANA (2001).

5) Vgl. WOLF (2001), S. 74 ff. vgl. WOLF (2001), S. 74 ff. Aus diesem Blickwinkel passt die Charakterisierung GLASERFELDs, der radikale Konstruktivismus sei eine „Erkenntnistheorie ohne Ontologie“; GLASERFELD (1987), S. 411.

gig existierenden Realität ist widersprüchlich zu der Annahme einer objektiven Erkennbarkeit der Realität.

Von den vorgestellten Positionierungen wird für die vorliegende Arbeit die Entscheidung zugunsten der Kombination aus ontologischem Realismus und erkenntnistheoretischem Idealismus getroffen. Zum einen geht der Verfasser davon aus, dass eine subjektunabhängige Realität existiert. Hierdurch wird die Perspektive eines ontologischen Realismus eingenommen. Zum anderen scheint die objektive Erkennbarkeit dieser Realität an verschiedenen Faktoren zu scheitern.[1] Entsprechend wird aus epistemologischer Perspektive eine idealistische Position eingenommen. Mit diesen wissenschaftstheoretischen Basisentscheidungen wird in der vorliegenden Arbeit eine *gemäßigt konstruktivistische* Position eingenommen.

Durch eine solche Positionierung werden mehrere konzeptionelle Probleme vermieden, die sich im Umgang mit Ontologien – im informationstechnischen Sinne – ergeben können. Erstens wird durch diese Positionierung nicht der Anspruch erhoben, Ontologien würden einen objektiven Realweltbezug haben. Eine solche Positionierung kann nur mit einem erkenntnistheoretischen Realismus in Einklang gebracht werden. Entsprechend der aufgeklärten idealistischen Epistemologie zufolge sind Ontologien Ergebnisse einer *Konstruktionsleistung*. Zwar wird die *Existenz* einer objektiven Realität nicht in Frage gestellt, jedoch die subjektunabhängige Erfahrbarkeit dieser objektiven Realität. Die Realität, auf die sich eine Ontologie bezieht, ist demnach stets das Ergebnis einer kognitiven Vorstrukturierung. Diese epistemologische Prämisse ist für die folgenden Ausführungen bedeutend, da sie nicht von allen Ausarbeitungen zu Ontologien zugrunde gelegt wird.[2]

Bezüglich der subjektbezogenen Konstruktion von Ontologien wird für die weiteren Ausführungen auf den *linguistischen* Aspekt ein Schwerpunkt gelegt.[3] Diesbezüglich wird davon ausgegangen, dass die Konstruktion von Erkenntnissen stets mit Hilfe einer *Sprache* erfolgt. Die Subjektbezogenheit der Erkenntnis, von der der erkenntnistheoretische Idealismus ausgeht, wird u.a. auf die Sprache zurückgeführt, mit der die Erkenntnis formuliert ist. Sprachliche Artefakte werden – entsprechend dieser Position – nicht als Abbilder eines Realitätsausschnitts aufgefasst, sondern als ihre sprachliche Rekonstruktion. Insofern ergeben sich zwei Determinanten für die Konstruktion von Erkenntnissen. Zum einen herrscht eine *Subjektabhängigkeit* in der Form vor, dass Individuen die Realität unterschiedlich erkennen. Zum anderen liegt eine *Sprachabhängigkeit* in der Form vor, dass zum Erkennen der Realität Individuen sich derjenigen sprachlichen Aus-

---

1) Für einen Katalog an Faktoren, die gegen einen erkenntnistheoretischen Realismus sprechen vgl. SCHÜTTE (1999), S. 226 f.

2) Vgl. hierzu die kritischen Anmerkungen in ZELEWSKI (2005). Rühmliche Ausnahmen hiervon sind die bereits angesprochenen Arbeiten GUARINOS, der auch eine Unterscheidung zwischen Ontologie und Erkenntnistheorie einschlägt; vgl. GUARINO (1995), S. 628 ff.

3) Zur Bedeutung linguistischer Aspekte für Erkenntnisprozesse vgl. FRANK (1999), S. 133 ff.; WINOGRAD/FLORES (1999), S. 17 ff.; WOLF (2001), S. 98 ff.

drucksmittel bedienen, über die sie verfügen. Verfügt ein Individuum nicht über die entsprechenden Ausdrucksmittel, kann es Phänomene auch nicht erkennen. Bezogen auf Ontologien ist die Ausdrückbarkeit von Sachverhalten der Ausdrucksmächtigkeit der Sprache untergeordnet, die zur Konstruktion der Ontologie verwendet wird. Insofern determinieren die zur Verfügung stehenden sprachlichen Primitive die potenziellen Konzeptualisierungsmuster. Für die Konstruktion von Ontologien hat dieser Sachverhalt eine zweifache Bedeutung.

Aus einem ersten Blickwinkel sind Ontologien *Ergebnisse* der Überführung natürlichsprachlicher Vorstrukturierungen in formalsprachliche Artefakte. Je nachdem, was für eine Ausdrucksmächtigkeit die Sprache innehat, in der die Ontologie konstruiert ist, können unterschiedliche Arten von Ausdrucksmitteln berücksichtigt werden. Beispielsweise gelten *Konzepte* und *Relationen* als notwendige Bestandteile jeder Ontologie.[1)] Insofern muss die formale Sprache, die für die Konstruktion der Ontologie verwendet wird, Ausdrucksmittel für die Spezifikation von Konzepten und Relation zur Verfügung stellen. Darüber hinausgehende Komponenten von Ontologien, wie z.B. Regeln, können nur dann ausgedrückt werden, wenn die Sprache auch hierfür Ausdrucksmittel bereitstellt.

Aus einem zweiten Blickwinkel ist eine Ontologie ein *Instrument* zum Gewinnen von Erkenntnissen über einen Realitätsausschnitt. Diese zweite Perspektive hat mit der ersten Perspektive gemeinsam, dass der Fokus auf Begriffe gerichtet ist, anhand derer eine Konzeptualisierung spezifiziert werden kann. Während jedoch die erste Perspektive die Metasprache zum Untersuchungsgegenstand hat, mit der eine Ontologie spezifiziert werden kann, wird in der zweiten Perspektive die Objektsprache untersucht, mit der die Realität repräsentiert werden kann. Auch hierbei werden durch die sprachlichen Primitive – die nunmehr von der Ontologie selbst zur Verfügung gestellt werden – Konzeptualisierungsmuster determiniert. Im Gegensatz zu der ersten Perspektive ist jedoch die zweite Perspektive nicht auf die *abstrakte* oder *schematische* Begriffsstruktur ausgerichtet, mit der Erkenntnisse konstruiert werden können. Im Vordergrund dieser zweiten Perspektive stehen die *konkreten* Objekte. Dabei sind die beiden Perspektiven derart miteinander „verdrahtet“, dass die Spezifikation *konkreter* Objekte mit Hilfe *abstrakter* Begriffe erfolgt. Beispielsweise kann in einem ontologiegestützten Modell eine konkrete Beziehung zwischen zwei Informationssystemen nur dann berücksichtigt werden, wenn die zugrunde gelegte Ontologie auch Begriffe zum Ausdrücken solcher Beziehungen zur Verfügung stellt. Insofern werden durch eine Ontologie Grenzen für Erfahrungsmöglichkeiten gesetzt.

Für die vorliegende Arbeit ist die Perspektive der abstrakten Ebene im Vergleich zur Perspektive der konkreten Ebene bedeutender. Denn es geht in der Arbeit vordergründig darum, präzise festzulegen, welche Ausdrucksmöglichkeiten Ontologien grundsätzlich eingeräumt werden. Erst auf der Basis einer solchen präzisen Festlegung kann eine Erweiterung von Ontologien um eine operationale Semantik formuliert werden.

---

1) Vgl. GRUBER (1993), S. 199.

2.1.2 Systemtheoretischer Rahmen

Terminologischer[1)] Ausgangspunkt für die Konzeptualisierungen der vorliegenden Arbeit ist die *Systemtheorie*.[2)] Mit der Systemtheorie wird der terminologische Bezugsrahmen für das integrative Modellierungskonzept festgelegt. Sie ist als terminologischer Bezugsrahmen geeignet, da sie einen Begriffsapparat zur Verfügung stellt, anhand dessen eine systematische Konzeptualisierung durchgeführt werden kann. Für die Entfaltung des terminologischen Bezugsrahmens werden daher zunächst die Grundzüge der Systemtheorie skizziert, um im Anschluss die Verknüpfungen zu dem Modellierungskonzept aufzuzeigen.

Von der Systemtheorie wird ein begriffliches Strukturierungsparadigma vorgelegt, das insbesondere in den wirtschaftswissenschaftlichen Teildisziplinen *Betriebswirtschaftslehre* und *Wirtschaftsinformatik* auf breite Resonanz stößt. Im Rahmen der Betriebswirtschaftslehre ist es insbesondere die *Organisationstheorie*, in der das Spektrum systemtheoretischer Denkansätze in Anspruch genommen wird.[3)] An der Schnittfläche zwischen Betriebswirtschaftslehre und Wirtschaftsinformatik werden auch im *Operations Research* Methoden der modellgestützten Analyse von Systemen untersucht. Ebenso hat die Systemtheorie für die *Wirtschaftsinformatik* eine große Bedeutung. So wird die Systemtheorie teilweise als Basisposition diskutiert, die systemtheoretischen Konzepte einerseits der Informatik und andererseits der Wirtschaftswissenschaften miteinander in einem Gedankengerüst zu harmonisieren.[4)] Die Harmonisierung beschränkt sich allerdings bisweilen darauf, eine gemeinsame terminologische Basis festzulegen, von der ausgegangen werden kann. Zudem sind die Festlegungen in der Regel auf natürlichsprachliche Erläuterungen beschränkt.[5)] Daher können mit den Erläuterungen Defekte der Mehrdeutigkeit und Vagheit verbunden sein, denen natürlichsprachliche Aussagensammlungen oftmals ausgesetzt sind. Insbesondere sozialwissenschaftliche Ansätze zur Systemtheorie entziehen sich oftmals einer disziplinenübergreifenden Anpassung, wenn formale und somit unmissverständliche Explikationen der Ansätze vermisst werden.

Bezüglich der Konzeptualisierungsmuster, die in natürlich- oder formalsprachlicher Form vorliegen und in eine formalsprachliche Ontologie überführt werden, wird von der vorliegenden Arbeit eine systemtheoretisch fundierte Struktur vorausgesetzt. Um die grundsätzliche Vergleichbarkeit von mentalen Konstruktionsprozessen der Akteure zu bewahren, wird den Prozessen eine „systemtheoretische Schablone" aufgesetzt. Insofern wird nicht davon ausgegangen, dass die Realität systemartig strukturiert sein muss. Es

---

1) Die Attribute „terminologisch" und „begrifflich" werden synonym verwendet.

2) Vgl. Ropohl (1979), S. 54 ff.; von Bertalanffy (1973) S. 29 ff.; Weinberg (975) S. 51 ff.

3) Vgl. Schreyögg (2003), S. 91 ff.; Beisel (1996), S. 17 ff.

4) Vgl. Petkoff (1997), S. 250 ff. (mit besonderem Bezug zum betrieblichen Wissensmanagement).

5) Eine rühmliche Ausnahme hiervon stellt Patig (2001a), S. 39 ff. dar. Die Ausnahmeposition wird dadurch gerechtfertigt, dass eine *strukturalistische* – und somit formale – Rekonstruktion der gesetzesartigen Aussagen der *allgemeinen Systemtheorie* erfolgt.

wird lediglich vorausgesetzt, dass die Erkenntnisse bezüglich dieser Realität systemartig sind.

Als System wird ein Zwei-Tupel aus einer Menge von *Elementen* und einer Familie[1)] von *Relationen* zwischen den Elementen verstanden.[2)] Jede Relation umfasst jeweils Tupel von Elementen, für die die Relation Gültigkeit besitzt. Jedes Tupel aus einer Relation wird als *Beziehung* bezeichnet. Isolierte Elemente, die in keiner Beziehung vorkommen, sind in Systemen ausgeschlossen.

Systeme können hinsichtlich ihrer *Struktur* und ihres *Verhaltens* charakterisiert werden.[3)] Zwecks Beschreibung der Struktur von Systemen werden das *strukturale* und das *hierarchische* Systemkonzept herangezogen.[4)] Der Beschreibung des Verhaltens dient das *funktionale Systemkonzept*. Die Abbildung 2 gibt einen Überblick über die Systemkonzepte.

---

1) Die Bezeichnung „Familie" wird in der vorliegenden Arbeit als Kurzform für die Bezeichnung „Mengenfamilie" verwendet. Bei Mengenfamilien handelt es sich um Zusammenfassungen von Mengen zu einer Gesamtheit. Die einzelnen *Mitglieder* $X_1,...,X_n$ einer Mengenfamilie $Y=(X_1,...,X_n)$ lassen sich eindeutig durch ihre Indizes voneinander unterscheiden. Ausgeschlossen ist dabei nicht, dass zwei Mitglieder X_a und X_b mit $a \neq b$ die gleichen Elemente umfassen. Entsprechend darf $X_a=X_b$ für beliebige $a,b \in \{1,...,n\}$ und $a \neq b$ gelten. Für den „Zugriff" auf die einzelnen Mitglieder $X_1,...,X_n$ einer Mengenfamilie $Y=(X_1,...,X_n)$ wird zudem in der vorliegenden Arbeit die Schreibweise $X_1,...,X_n \in Y$ zugelassen.

   Da es sich auch bei Relationen um Mengen handelt – nämlich um Mengen von *Beziehungen* zwischen einzelnen Elementen – wird bei der Zusammenfassung von Relationen aus einem System auch von einer Familie gesprochen. Zwei Relationen $R_a \subseteq X_1 \times ... \times X_n$ und $R_b \subseteq X_1 \times ... \times X_n$ aus einem System können entsprechend durchaus die gleichen Beziehungen $(q_1,...,q_n),....,(r_1,...,r_n) \in X_1 \times ... \times X_n$ umfassen. In einer Familie $RF=\{...,R_a,R_b,...\}$ können die beiden Relationen R_a und R_b die gleichen Elemente enthalten, d.h. es darf auch $R_a=R_b$ gelten.

2) Vgl. BECKER (1999), S. 12; BOSSEL (1994), S. 16; FERSTL/SINZ (1998), S. 11; SCHÜTTE (1998), S. 37; SCHULZE (2001), S. 9; WOLF (2001), S. 13 (kritisch); ZELEWSKI (1995), Bd. 2, S. 49. Die vorgelegte Definition von Systemen umfasst auch die Eigenart von Systemen, *Eigenschaften* oder *Attribute* der einzelnen Elemente zu umfassen. Denn Eigenschaftsausprägungen oder Attributswerte von Elementen lassen sich als Beziehungen zwischen den entsprechenden Elementen auffassen. Auch bei den Eigenschaftsausprägungen handelt es sich um Elemente der Menge, die für das System vorausgesetzt wird.

3) Vgl. WOLF (2001), S. 14 ff.; ZELEWSKI (1995), Bd. 2, S. 49.

4) Vgl. ROPOHL (1979), S. 54 ff.

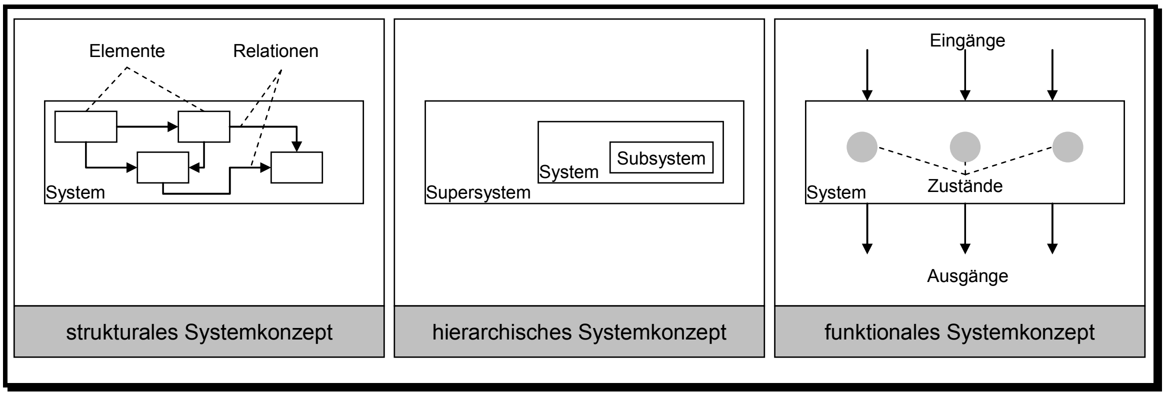

Abbildung 2: Systemkonzepte[1)]

Im *strukturalen Systemkonzept* wird ein System über die Definitionsmerkmale Elemente und Relationen identifiziert. Systeme unterscheiden sich demnach voneinander dadurch, welche Elemente in welcher spezifischen Anordnung zueinander stehen. Dem strukturalen Systemkonzept folgend, können Elemente nicht losgelöst von den Systemen, in die sie eingebettet sind, betrachtet werden. Jedes Element wird durch die Beziehungen charakterisiert, die es zu anderen Elementen unterhält. Somit wird im strukturalen Systemkonzept ein Ansatz angestrebt, mit dem Unterschiede zwischen Systemen basierend auf den Beziehungen zwischen Elementen ausgemacht werden können.

Der Betrachtungsgegenstand des strukturalen Systemkonzepts lässt sich auch als *Objektebene* charakterisieren. Im Gegensatz zum strukturalen Systemkonzept ist im hierarchischen Systemkonzept nicht die Objektebene Betrachtungsgegenstand, sondern eine *Metaebene*. Im Gegensatz zur Objektebene des strukturalen Systemkonzepts werden auf der Metaebene des hierarchischen Systemkonzepts nicht die Elemente und deren Beziehungen untereinander untersucht, sondern die Beziehungen von Systemen untereinander. Das hierarchische Systemkonzept erlaubt eine komplexitätsreduzierende Verschachtelung von Systemen durch eine Relation[2)], die nicht über der Menge der Elemente formuliert wird, sondern über einer Menge höherer Ordnung.[3)] Mit dem Übergang von einem hierarchisch untergeordneten zu einem hierarchisch übergeordneten System ist eine *Vergröberung* verbunden. Im umgekehrten Fall erfolgt eine *Verfeinerung*, wenn von einem hierarchisch übergeordneten System zu einem hierarchisch untergeordneten System übergegangen wird.

1) Vgl. ROPOHL (1979), S. 55.

2) Bei der Relation zur Formulierung hierarchischer Beziehungen zwischen Teilsystemen handelt es sich um eine *partielle Ordnungsrelation*. Partielle Ordnungsrelationen werden in der vorliegenden Arbeit mehrfach vorgestellt und an den jeweiligen Stellen auch formal präzisiert.

3) Vgl. ROPOHL (1979), S. 60. Dieser Konzeptualisierung wird in der vorliegenden Arbeit dadurch gefolgt, dass Relationen zur Formulierung von Unter- und Überordnungsbeziehungen zwischen Konzepten aus Ontologien durch *metasprachliche* Ausdrücke festgelegt werden.

Sowohl das strukturale als auch das hierarchische Systemkonzept erlauben lediglich zustandsinvariante Aussagen über Systeme. Zwischen verschiedenen Zuständen eines Systems wird weder im strukturalen noch im hierarchischen Systemkonzept unterschieden. Diese Perspektive wird durch das *funktionale Systemkonzept* eingenommen. Hierbei erfolgt eine verhaltensorientierte Charakterisierung von Systemen, indem sie anhand ihrer Ein- und Ausgänge beschrieben werden, durch die sie mit ihrer Systemumwelt interagieren.[1)] Mit den Interaktionsmöglichkeiten eines Systems wird festgelegt, welche Zustände für ein System zulässig sind. Die Menge aller Zustände, die ein System einnehmen kann, wird als *Zustandsraum* bezeichnet. Eine zweistellige Relation über dem Zustandsraum umfasst alle *Zustandsübergänge*. Wenn es sich um eine rechtseindeutige Relation – also eine Operation[2)] – handelt, wird das System als *deterministisch* bezeichnet. Ansonsten handelt es sich um ein *nicht-deterministisches* System.

Das funktionale Systemkonzept wird von dem strukturalen und dem hierarchischen Systemkonzept dadurch abgegrenzt, dass die inneren Zusammenhänge eines Systems, die zur Transformation des Systemzustandes führen, bei dem erstgenannten Systemkonzept nicht von Relevanz sind. Es wird lediglich betrachtet, wie ein System auf bestimmte Einflüsse aus der Systemumwelt reagiert. Insofern liegt eine „black box"-Betrachtung des Systems vor.[3)] Für das strukturale und das hierarchische Systemkonzept sind hingegen innere Zusammenhänge des Systems von Relevanz. Das System wird einer „glass box"-Betrachtung unterzogen, indem entweder das Beziehungsgeflecht zwischen den Elementen oder das Beziehungsgeflecht zwischen Systemen analysiert werden. Im zweiten Fall werden die einzelnen Systeme weiterhin als „black boxes" betrachtet. Lediglich die Beziehungen zwischen den Systemen sind erkennbar.

Der systemtheoretischen Sichtweise wird in der vorliegenden Arbeit aus verschiedenen Perspektiven Rechnung getragen. Beispielsweise werden *signaturspezifische Strukturen* als formale Konstrukte zur Repräsentation von Systemen verwendet. Dabei lassen sich die signaturspezifischen Strukturen selbst als Systeme im oben angegebenen Sinn auf-

---

1) Die verhaltensorientierte Charakterisierung von Systemen ist im Besonderen für die *Kybernetik* von Bedeutung. ASHBY bringt den Zusammenhang mit der Aussage: „Er (der Kybernetiker [Anm. d. Verf.]) geht grundlegend funktional und behaviouristisch vor." (ASHBY (1974), S. 15) zum Ausdruck.

2) Die Bezeichnungen *Operation* und *Funktion* werden im Folgenden synonym verwendet. Es handelt sich dabei stets um rechtseindeutige Relationen.

3) Vgl. BOSSEL (1994), S. 29; MITTELMANN (1995), S. 14.

fassen.[1)] Der Realitätsausschnitt, der durch die Struktur repräsentiert wird, stellt hierbei ein *Objektsystem* dar. Die Struktur ist in diesem Fall ein *Modellsystem*.[2)]

Darüber hinaus ist die systemtheoretische Konzeptualisierung mit einer *objektorientierten* Sichtweise vereinbar.[3)] Dabei ist die objektorientierte Sichtweise dem Prinzip signaturspezifischer Strukturen unmittelbar behaftet. Grundsätzlich könnte hierbei der Begriff des *Elementes*, der bei der Definition von Systemen im Vordergrund steht, durch den Begriff des *Objektes* ersetzt werden. Bei näherer Betrachtung erweist sich der Objektbegriff jedoch als reichhaltiger als der Elementbegriff. Für Objekte wird gemeinhin die Möglichkeit einer inneren Struktur eingeräumt, die in der Regel für Elemente ausgeschlossen ist. Als Elemente werden in der Systemtheorie atomare Einheiten von Systemen verstanden, die keine innere Struktur aufweisen können.[4)] Objekte können hingegen sowohl atomarer als auch zusammengesetzter Natur sein. Als atomare Objekte werden im Folgenden solche Einheiten verstanden, die keine innere Struktur aufweisen. Sie stimmen mit dem üblichen Elementverständnis der Systemtheorie überein. Für komplexe Objekte ist eine solche innere Struktur hingegen zulässig. Sie werden als Teilsysteme eines Systems konzeptualisiert. Als Grenzfälle von Teilsystemen werden Elemente zugelassen.

In der vorliegenden Arbeit gehen komplexe Objekte aus der Anwendung von *Operationen* auf andere – weniger komplexe – Objekte hervor. Bei den Objekten, auf die Operationen angewandt werden, kann es sich wiederum um komplexe Objekte handeln, die selbst aus der Anwendung einer Operation auf Objekte hervorgegangen sind. Ihre Komplexität ist jedoch immer geringer als die Komplexität des Objektes, das mit ihrer Hilfe generiert wird, da mit der Anwendung einer Operation auf Objekte eine weitere Komplexitätsstufe erreicht wird. Somit steigt mit zunehmend *verschachtelter* innerer Struktur eines Objektes auch dessen Komplexität. Bei der komplexitätsgenerierenden Verschachtelung der inneren Struktur komplexer Objekte wird zu ihrer Definition das *Induktionsprinzip* herangezogen. Als Induktionsbasis müssen für die Konstruktion komplexer Objekte atomare formale Objekte verwendet werden.

In einer weiteren Differenzierung können Objekte aus einer tatsächlichen oder gedachten Realität von Objekten unterschieden werden, die lediglich sprachlicher Natur sind und in einem formalen Kalkül Verwendung finden. Erstgenannte werden im Folgenden als *reale Objekte* bezeichnet. Zweitgenannte werden als *formale Objekte* bezeichnet.

---

1) Vgl. SCHULZE (2001), S. 11 ff. Als signaturspezifische Strukturen werden in der vorliegenden Arbeit Strukturen zu konventionellen (vgl. Abschnitt 2.2.1.1.3.1), sortierten (vgl. Abschnitt 2.2.1.2.3.1) und ontologischen Signaturen (vgl. Abschnitt 3.1.3.1.1) vorgestellt. Sämtliche Strukturen umfassen jeweils eine Menge von Elementen und Familien von Relationen. Insofern können sie als Systeme in dem hier vorgelegten Verständnis aufgefasst werden.

2) Zu den Begriffen *Objekt-* und *Modellsystem* vgl. FERSTL/SINZ (1998), S. 18.

3) Vgl. BECKER (1999), S. 76 ff.; ZELEWSKI (1995), Bd. 2, S. 55 ff.

4) Vgl. BECKER (1999), S. 13.

2.1.3 Modelltheoretischer Rahmen

2.1.3.1 Modellierung

Zur Handhabung der Komplexität von Systemen werden in der vorliegenden Arbeit *formale Modelle* diskutiert.[1)] Bei formalen Modellen handelt es sich selbst um (Modell-) Systeme. Die Beschäftigung mit Modellen wird dadurch motiviert, dass es sich bei Modellen um komplexitätsreduzierende Instrumente zur Handhabung von realen oder gedachten Systemen handelt. Um einen terminologischen Bezugsrahmen für die Untersuchung von Modellen zu erhalten, wird zunächst die allgemeine Modelltheorie STACHOWIAKs herangezogen. Demnach können Modelle anhand dreier Merkmale charakterisiert werden:[2)]

- Abbildungsmerkmal
  Diesem ersten Merkmal zufolge sind Modelle stets *Abbildungen* von natürlichen oder künstlichen *Originalen*. Bei den Originalen kann es sich wiederum um Modelle handeln.[3)]
- Verkürzungsmerkmal
  In Modellen werden nicht alle Eigenschaften des Originalsystems erfasst, sondern nur solche, die für den Modellierungskontext relevant[4)] sind. Merkmale des Originals, die im Modell nicht enthalten sind, werden im Rahmen der modellierungsspezifischen *Präterition* fallengelassen.[5)] Im Gegenzug können Modelle über so genannte *abundante* Merkmale verfügen, die das entsprechende Original nicht aufweist.
- Pragmatisches Merkmal
  Modelle ersetzen ihre Originale für Subjekte innerhalb bestimmter Zeitintervalle zu einem bestimmten Zweck.[6)] Insofern sind Modelle subjekt-, zeit- und zweckbezogen.

Für die allgemeine Modelltheorie STACHOWIAKs ist das Konzept der *semantischen Stufen* von äußerster Bedeutung.[7)] Mit dessen Hilfe lassen sich Prozesse der Konstruktion

---

1) Dabei wird zunächst offen gelassen, ob das Original, das der Modellierung zugrunde liegt, einen Realitäts*bezug* hat oder sogar die Realität *ist*. Auf den Realitätsbezug von Modellen wird in den folgenden Anmerkungen näher eingegangen.

2) Vgl. STACHOWIAK (1973), S. 131 ff.

3) Vgl. STACHOWIAK (1983), S. 118.

4) Die Relevanz der Eigenschaften steht in unmittelbarem Zusammenhang mit dem pragmatischen Merkmal von Modellen, das als Nächstes angesprochen wird; vgl. SCHÜTTE (1998), S. 41; WOLF (2001), S. 66.

5) Vgl. WOLF (2001), S. 65.

6) Vgl. STACHOWIAK (1973), S. 132 f.; STACHOWIAK (1983), S. 122 ff.

7) Vgl. STACHOWIAK (1973), S. 196 ff.

von Modellen in einem semiotischen Bezugsrahmen systematisch beschreiben. Ein solcher semiotischer Bezugsrahmen wird in der vorliegenden Arbeit mehrfach in Anspruch genommen. Beispielsweise erweist sich das Konzept der semantischen Stufen als nützlich, um das Prinzip der *denotationalen Interpretation*[1)] von sprachlichen Konstrukten in Ontologien – auf das in späteren Abschnitten ausführlich zurückgekommen wird[2)] – zu verdeutlichen. Für diese Zwecke reicht es im Folgenden aus, auf jene Aspekte des Konzepts semantischer Stufen einzugehen, die für die vorliegende Arbeit von Relevanz sind.[3)]

Die *nullte* oder *uneigentliche* Stufe enthält Klassen von *Taxen*. Es handelt sich bei Taxen um „materiell-energetische" Einheiten, „die Originalkonfigurationen für Perzeptionsmodelle [...] liefern"[4)]. Jede Klasse von Taxen konstituiert ein *Taxem*. Taxeme sind atomare Ausdruckseinheiten, mit denen Ausdrücke konstruiert werden können. Die Bedeutungen der Ausdrücke sind auf dieser Stufe nicht von Relevanz.

Auf der *ersten* semantischen Stufe sind die *Perzeptionsmodelle* und die *kogitativen Modelle* von Akteuren angesiedelt. In ihrer Gesamtheit entsprechen Perzeptionsmodelle und kogitative Modelle den *internen Modellen* von Akteuren. In den Perzeptionsmodellen spiegelt sich die Wahrnehmung des Akteurs bezüglich eines Originals wider. Kogitative Modelle gehen aus der Kombination und Umstrukturierung von Perzeptionsmodellen hervor.

Interne Modelle von Akteuren sind die Ergebnisse von introvertierten Kognitionsleistungen und sind daher für die Kommunikation mit anderen Akteuren nicht geeignet.[5)] Modelle, die aus der Übertragung interner Modelle in eine kommunizierbare Form hervorgehen, sind auf der *zweiten* semantischen Stufe angesiedelt. Somit sind Modelle auf der zweiten Stufe stets Modelle von Originalen der ersten Stufe.[6)]

Auf der *dritten* semantischen Stufe schließlich sind Modelle von Originalen der zweiten Stufe enthalten. Gegenstand von Modellen der dritten Stufe sind die Zeichen, die zur Konstruktion von Modellen der zweiten Stufe verwendet werden können. Modelle der dritten Stufe sind – wie Modelle der zweiten Stufe auch – im Gegensatz zu Modellen der ersten Stufe zur Kommunikation zwischen Akteuren geeignet.

---

1) Die denotationale Interpretation eines sprachlichen Konstrukts ist die Zuordnung eines realweltlichen, außersprachlichen Konstrukts; vgl. ZELEWSKI (1995), Bd. 2, S. 350). Die denotationale Semantik eines sprachlichen Konstrukts entspricht diesem realweltlichen Pendant.

2) Vgl. Abschnitt 3.1.3.2

3) Zu ähnlich verkürzten Darstellungen des Konzept semantischer Stufen vgl. SCHÜTTE (1998), S. 42 ff. und WOLF (2001), S. 66 ff.

4) STACHOWIAK (1983), S. 124.

5) Vgl. RAPAPORT (2003), S. 403.

6) Vgl. STACHOWIAK (1983), S. 123.

Über die dritte Stufe hinaus kann das Konzept der semantischen Stufen weitergeführt werden. Mit den Zeichen der Stufe n mit n>0 werden jeweils Modelle der Stufe n-1 konstruiert.[1)] Die Gesamtheit von Zeichen und Regeln für die Konstruktion von Ausdrücken, konstituiert eine *Modellierungssprache*.[2)]

Auf dem Konzept der semantischen Stufen basierend können *Objekt-* und *Metasprachen* voneinander getrennt werden.[3)] Eine Objektsprache umfasst die Gesamtheit aller Zeichen und Regeln, mit denen *Objektmodelle* konstruiert werden können. Metasprachen umfassen hingegen Zeichen und Konstruktionsregeln für metasprachliche Ausdrücke, die über Objektmodelle getroffen werden können. Insofern sind die Eigenarten einer Sprache, eine Objekt- oder eine Metasprache zu sein, immer in Bezug zu einer weiteren Meta- bzw. Objektsprache gegeben. Weiterführend können über eine Metasprache mit Hilfe einer Meta-Metasprache Ausdrücke konstruiert werden. Die rekursive Verschachtelung von Sprachen kann beliebig fortgeführt werden.

Für die inhaltliche Festlegung, welches Verständnis mit dem Begriff *Modell* verbunden wird, bieten sich unterschiedliche Auffassungen an. Die unterschiedlichen Modellbegriffe basieren zumeist auf unterschiedlichen wissenschaftstheoretischen Ausrichtungen. In den Wirtschaftswissenschaften ist zumeist der abbildungsorientierte Modellbegriff am weitesten verbreitet. Dem abbildungsorientierten Modellbegriff zufolge sind Modelle Abbilder einer sowohl objektiv existierenden als auch objektiv erfahrbaren Realität.[4)] Während es die allgemeine Modelltheorie noch offen lässt, in was für einem Verhältnis die betrachteten Originale zur Realität stehen, wird im abbildungsorientierten Verständnis für Modelle das Original mit der Realität gleichgesetzt. Dabei wird die Abbildungsbeziehung zwischen der Realität und dem Modell oftmals als *isomorphe* oder als *homomorphe* Beziehung charakterisiert.[5)] Die Abbildungsbeziehung lässt sich im Fall der Isomorphie als eineindeutige Funktion charakterisieren, durch die eine Strukturidentität von Modell und Realität bewahrt wird. Eine Isomorphieforderung wird allerdings in der Regel auch von Anhängern abbildungsorientierter Modellbegriffe nicht erhoben. Das liegt zum einen daran, dass der Begriff der Isomorphie als mathematische Funktion zwischen Strukturen definiert ist,[6)] der sich nicht ohne weiteres auf eine Beziehung zwischen Realität und Modell übertragen lässt. Zum anderen würde mit einer isomorphen Modellierung eine „Weltverdoppelung“[7)] einhergehen, die dem Verkür-

---

1) Vgl. SCHÜTTE (1998), S. 45.

2) Zu den Beziehung zwischen Modellierungssprachen und Modellen vgl. LEHNER ET AL. (1995), S. 81 ff.; WOLF (2001), S. 53 ff.

3) Zur Unterscheidung zwischen *Objekt-* und *Metasprache* vgl. STEIMANN (2000), S. 18 f.; ROTHMALER (1995), S. 29; WEDEKIND ET AL. (2004A), S. 459 ff.

4) Vgl. WOLF (2001), S. 52.

5) Vgl. FERSTL/SINZ (1998), S. 19.

6) Vgl. EHRIG ET AL. (1999), S. 144; ROTHMALER (1995), S. 21 ff.

7) ZELEWSKI (1995), Bd. 2, S. 23.

zungsmerkmal aller Modelle widerspräche. Auch der Begriff der Homomorphie ist als mathematische Funktion definiert.[1] Im Kontext abbildungsorientierter Modellbegriffe wird der Begriff der Homomorphie im „bildlichen Sinne“[2] als eine solche Abbildung charakterisiert, durch die die Struktur-Ähnlichkeit von Modell und Realität bewahrt wird.

Dem abbildungsorientierten Modellbegriff zufolge werden die Prozesse zur Konstruktion von Modellen darauf reduziert, die Realität zu erfassen und abzubilden. An den Modellierer werden somit lediglich die Anforderung für „ein geschultes Auge“, „eine Aufnehmungsgabe für die Realität“[3] und die Beherrschung der Modellierungssprache gestellt.[4] Diese Sichtweise steht allerdings im Widerspruch zu der wissenschaftstheoretischen Basisposition, die für die vorliegende Arbeit vorausgesetzt wurde. Dort wurden Auffassungen zurückgewiesen, denen zufolge nicht nur ein ontologischer, sondern auch ein epistemologischer Realismus vertreten wird. Während der ontologische Realismus für die vorliegende Arbeit vorausgesetzt wurde, wurde der mit dem abbildungsorientierten Modellbegriff einhergehende epistemologische Realismus[5] abgelehnt. Daher wird der vorliegenden Arbeit der abbildungsorientierte Modellbegriff nicht zugrunde gelegt.

Für die vorliegende Arbeit wird der *konstruktionsorientierte* Modellbegriff[6] SCHÜTTES vorausgesetzt, der auf der Differenzierung zwischen Ontologie – im philosophischen Sinne – und Epistemologie basiert. SCHÜTTE folgend kann ein *Modell* als „das Ergebnis einer *Konstruktion eines Modellierers*, der für *Modellnutzer* eine Repräsentation eines Originals zu einer Zeit als relevant mit Hilfe einer *Sprache* deklariert“[7] definiert werden. Mit der Übernahme des konstruktionsorientierten Modellbegriffs sind für die folgenden Ausführungen einige Konsequenzen verbunden. Im Mittelpunkt der Definition steht die *Konstruktionsleistung* eines Modellierers. Dadurch wird explizit auf eine konstruktivistische Sichtweise Bezug genommen. Modelle werden keinesfalls als Abbildungen aufgefasst, deren Ähnlichkeit mit der Realität beurteilt werden könnte. Modelle

---

1) Vgl. EHRIG ET AL. (1999), S. 136 ff.; ROTHMALER (1995), S. 20.

2) SCHÜTTE (1998), S. 55; ähnlich in ZELEWSKI (1995), Bd. 2, S. 23 („metaphorisch“).

3) DRESBACH (1999), S. 74.

4) Vgl. SCHÜTTE (1998), S. 48.

5) Dass der epistemologische Realismus mit dem abbildungsorientierten Modellbegriff einhergeht, ergibt sich daraus, dass die Prüfbarkeit der Abbildungsbeziehung unterstellt wird. Würde von einer subjektabhängigen Erfahrbarkeit der Realität ausgegangen werden, könnte die Abbildungsbeziehung zwischen Modell und Realität auch nicht überprüft werden; vgl. SCHÜTTE (1998), S. 56 f.; ZELEWSKI (1995), Bd. 2, S. 23. Insofern muss jeder abbildungsorientierte Modellbegriff auch mit der Annahme einer subjektunabhängigen Erfahrbarkeit der Realität, also mit der Position des epistemologischen Realismus, einhergehen.

6) Zu konstruktionsorientierten Modellbegriffen vgl. SCHÜTTE (1998), S. 59 ff.; SCHULZE (2001), S. 19 ff.; WOLF (2001), S. 74 ff.

7) SCHÜTTE (1998), S. 59.

repräsentieren vielmehr die Wahrnehmung eines Akteurs bezüglich einer Realität. Insofern weisen Modelle stets eine subjektive Komponente auf. Jedes Modell ist darauf bezogen, wie der modellierende Akteur den modellierten Realitätsausschnitt wahrgenommen hat.

Die Konstruktion von Modellen erfolgt stets unter Rückgriff auf eine Sprache als Instrument der Modellierung. Damit der Modellierer seine Wahrnehmungen bezüglich eines Realitätsausschnitts gegenüber anderen kommunizieren kann, muss er sich einer Sprache bedienen, die auch von den Adressaten verstanden wird. Dies entspricht dem Übergang von der ersten zur zweiten semantischen Stufe.

Werden zur Modellierung nur sprachliche Ausdrucksmittel verwendet, die in einer Ontologie spezifiziert sind, handelt es sich um ein *ontologiegestütztes Modell*. Ein ontologiegestütztes Modell setzt sich aus einer Ontologie und *Fakten* zusammen. Jedes Faktum ist eine wahrheitsdefinite Formel, mit der über einen Realitätsausschnit eine Aussage getroffen wird. Jedes Faktum wird mit den sprachlichen Ausdrucksmitteln aus der Ontologie konstruiert. Insofern stellen Ontologien u.a. einen Vorrat an Ausdrucksmitteln zur Verfügung, anhand derer Modelle der zweiten Stufe konstruiert werden können. Die Funktionalität von Ontologien ist jedoch nicht darauf beschränkt, lediglich einen syntaktischen Zeichenvorrat zur Verfügung zu stellen. Auch die *Bedeutung* ontologiegestützter Modelle ist eindeutig. Dies wird durch die formal festgelegte Bedeutung der sprachlichen Ausdrucksmittel gewährleistet. Die Bedeutungen der sprachlichen Ausdrucksmittel, die für die Modellierung verwendet werden, sind somit nicht missverständlich. Darüber hinaus verfügen Ontologien über metasprachliche Ausdrucksmittel, anhand derer die Bedeutungen der sprachlichen Ausdrucksmittel auch natürlichsprachlich ausgedrückt werden können.

2.1.3.2 Anforderungen an das Modellierungskonzept

2.1.3.2.1 Überblick über den Anforderungskatalog

Um die Eignung des integrativen Modellierungskonzepts beurteilen zu können, wird ein *Anforderungskatalog* vorgestellt. Der Anforderungskatalog umfasst *Anforderungen*[1)] an Ansätze zur Erweiterung von Ontologie-Sprachen um dynamische Aspekte. Er wird zu einer späteren Evaluation des integrativen Modellierungskonzepts herangezogen. Dazu umfasst der Anforderungskatalog sowohl *generische* Anforderungen, die für Modellierungskonzepte im Allgemeinen gelten, als auch Anforderungen, die sich aus dem *speziellen* Umstand der Modellierung kooperativer Informationssysteme ergeben.

Entsprechend der Zielsetzung der vorliegenden Arbeit, Ontologien in ein ganzheitliches Modellierungskonzept einzubinden, durch das sowohl statische als auch dynamische Aspekte berücksichtigt werden können, werden zwei Anforderungskataloge vorgestellt. Von dem ersten Anforderungskatalog werden Anforderungen zur Beurteilung der Güte

1) Die Begriffe *Anforderung* und *Kriterium* werden im Folgenden synonym verwendet.

des integrativen Modellierungskonzepts bezüglich der Modellierbarkeit *statischer* Phänomene umfasst. Die Ausdrucksmöglichkeiten des integrativen Modellierungskonzepts werden durch die Beurteilung seiner *statischen Modellierungsfähigkeit* bestimmt.[1)]

Zur Modellierung der statischen Aspekte werden für das integrative Modellierungskonzept Ontologien vorausgesetzt. Entsprechend kann die statische Ausdrucksmächtigkeit des Modellierungskonzepts mit der Ausdrucksmächtigkeit der Ontologie-Sprache gleichgesetzt werden, die vorausgesetzt wird. Um die *relative* Vorteilhaftigkeit des integrativen Modellierungskonzepts bestimmen zu können, wird einerseits der Anforderungsteilkatalog für die statische Modellierungsfähigkeit im Rahmen der späteren Evaluation zusätzlich auf alternative Ontologie-Sprachen angewandt. Zudem wird eine *Referenzontologie* in sämtlichen vorgestellten Ontologie-Sprachen rekonstruiert. Dazu wird die Referenzontologie in der eigens entwickelten Ontologie-Sprache konstruiert, um sie im Anschluss in den alternativen Ontologie-Sprachen zu rekonstruieren.[2)]

Analog zur Beurteilung seiner statischen Modellierungsfähigkeit wird die Ausdrucksmächtigkeit des Modellierungskonzepts bezüglich dynamischer Phänomene durch die Beurteilung seiner *dynamischen Modellierungsfähigkeit* bestimmt. Im Gegensatz zur Beurteilung der statischen Modellierungsfähigkeit erfolgt jedoch hinsichtlich der Beurteilung der dynamischen Modellierungsfähigkeit keine *konzeptexogene* Evaluation. In die Beurteilung der dynamischen Modellierungsfähigkeit werden nämlich keine Alternativen einbezogen, mit denen Ontologie-Sprachen um dynamische Aspekte angereichert werden könnten. Insofern handelt es sich bei der Beurteilung der dynamischen Modellierungsfähigkeit um eine *konzeptendogene* Evaluation.[3)]

---

1) Es wird ausdrücklich darauf hingewiesen, dass der Begriff „Modellierungsfähigkeit" nicht auf die Konstruktion von Ontologien bezogen ist, sondern auf die Konstruktion von Modellen mit Hilfe von Ontologien. Solche Modelle wurden bereits als ontologiegestützte Modelle vorgestellt.

2) Der Verfasser ist sich bewusst, dass eine solche sequentielle Rekonstruktion der Referenzontologie nicht dazu ausreichen kann, die Ausdrucksmächtigkeit aller Ontologie-Sprachen bewerten zu können, da sich in diesem Fall nur solche Ausdrucksmöglichkeiten der alternativen Ontologie-Sprachen demonstrieren lassen, die bereits bei der initialen Konstruktion der Referenzontologie ausgeschöpft wurden.

3) Die Beschränkung auf eine konzeptendogene Evaluation erfolgt vornehmlich aus zwei Gründen. Zum einen wird auf einen Vergleich des Ansatzes mit alternativen Modellierungskonzepten verzichtet, da möglicherweise unterschiedliche Zwecksetzungen vorliegen könnten. Die Zwecksetzung des integrativen Modellierungskonzepts ist darauf beschränkt, Zustandsübergänge in Modellen kooperativer Informationssysteme beschreiben zu können, die ontologiebasiert konstruiert wurden. Unterschiedliche Zwecksetzungen alternativer Konzepte könnten zu einer „Verzerrung" der Evaluation führen. Zum anderen sind dem Verfasser bislang keine Arbeiten bekannt, in denen die Einbettung von Ontologien in ein integratives Modellierungskonzept, das auch dynamische Phänomene zu erfassen in der Lage ist, behandelt würde.

2.1.3.2.2 Anforderungskatalog für die statische Modellierungsfähigkeit

Die statische Modellierungsfähigkeit des Modellierungskonzepts wird anhand von Anforderungen beurteilt, anhand derer die *Ausdrucksmächtigkeit* einer Ontologie-Sprache beurteilt werden kann. Dabei wird die Ausdrucksmächtigkeit anhand derjenigen metasprachlichen Ausdrucksmittel bemessen, über die eine Ontologie-Sprache verfügt und mittels derer Ausdrücke über objektsprachliche Ausdrucksmittel konstruiert werden können.

Bisweilen wurden bereits mehrere Kataloge von Anforderungen vorgestellt, die für die Beurteilung der Ausdrucksmächtigkeiten von Ontologie-Sprachen in Frage kommen.[1)] Der folgende Anforderungskatalog stellt einerseits ein Exzerpt aus diesen Arbeiten dar. Andererseits umfasst der Katalog auch solche Anforderungen, die bislang keine Berücksichtigung gefunden haben.

Der Anforderungskatalog kann – den Bestandteilen von Ontologien entsprechend – in drei *Teilkataloge* aufgeteilt werden. Bei den hinreichenden Komponenten handelt es sich um *Konzepte*, *Relationen* und *Regeln*.[2)] Im ersten Teilkatalog sind solche Anforderungen enthalten, die die Spezifikationsmittel der Ontologie-Sprachen für Konzepte betreffen. Im zweiten Teilkatalog sind Anforderungen enthalten, mit denen die Spezifikationsmittel für *Relationen*, mit denen einzelne *Beziehungen* zwischen Instanzen von Konzepten ausgedrückt werden können, untersucht werden können. Anhand des dritten Teilkatalogs können schließlich Mittel zur Spezifikation *regelartiger Zusammenhänge* untersucht werden.

❒ Taxonomie

Bei der ersten Anforderung handelt es sich darum, ob Konzepte in der jeweiligen Sprache zueinander in eine *taxonomische Beziehung* gesetzt werden können. Die jeweilige Sprache erlaubt genau dann die Spezifikation von taxonomischen Beziehungen zwischen Konzepten, wenn Konzepte als *Subkonzepte* anderer Konzepte ausgezeichnet werden können. Dabei wird ein erstes Konzept genau dann als *Subkonzept* eines zweiten

---

1) Vgl. CORCHO/GOMEZ-PEREZ (2000), S. 81 ff.; GOMEZ-PEREZ ET AL. (2004), S. 199 ff.; STUCKENSCHMIDT (2003), S. 33 ff. Vgl. darüber hinaus FENSEL (2000), S. 111 ff. für einen unstrukturierten Vergleich von Ontologie-Sprachen. Für einen allgemeiner Rahmen zur Evaluation von Sprachen zur konzeptionellen Modellierung vgl. PATIG (2004), S. 97 ff.

2) Vgl. GRUBER (1993), S. 199; SURE (2003), S. 22 f.

Konzepts verstanden, wenn die Extension[1)] des zweiten Konzepts alle Objekte aus der Extension des ersten Konzepts immer umfassen muss.

Konzept-Taxonomien werden aufgrund ihrer Bedeutung für Ontologien des Öfteren auch als das „Rückgrat“ einer Ontologie bezeichnet.[2)] Diese Bedeutung von Taxonomien für Ontologien hat unterschiedliche Gründe. Beispielsweise ist die Unterordnung von Strukturierungseinheiten ein Verfahren, das von nahezu allen Modellierungsansätzen unterstützt wird. Sowohl von traditionellen Sprachen zur konzeptionellen Modellierung, wie z.B. der *Entity-Relationship-Modellierung*[3)], als auch von aktuell diskutierten Sprachen, wie z.B. der *Unified Modeling Language*[4)] (UML), werden Ausdrucksmittel zur Verfügung gestellt, um die jeweils verwendeten Strukturierungseinheiten zueinander in taxonomische Beziehungen setzen zu können. Aus Gründen der Anschlussfähigkeit der ontologiegestützten Modellierung an Methoden zur konzeptionellen Modellierung ist die Spezifizierbarkeit von Konzept-Taxonomien durch Ontologie-Sprachen wünschenswert. Darüber hinaus ist die Generalisierung bzw. Spezialisierung von Strukturierungseinheiten ein Instrument der *Komplexitätsreduktion*.[5)] Zum einen wird bei der Generalisierung von Modellierungseinheiten bei einer zweckorientierten Analyse der Modelle eine Reduktion auf zweckrelevante Aspekte berücksichtigt. Zum anderen wird bei der Spezialisierung die für eine zweckrelevante Analyse zu grobe Modellsicht um verfeinerte Aspekte bereichert.[6)] Der Verfeinerung der Modellsicht erfolgt durch die Definition von Eigenschaften, die für hierarchisch übergelagerte Strukturierungseinheiten nicht definiert sind.

Wenn eine Sprache die Spezifikation taxonomischer Beziehungen zwischen Konzepten unterstützt, sollte sie Ausdrucksmöglichkeiten, die für ein Konzept festgelegt wurden,

---

1) Die *Extension* eines Konzepts ist dieMenge der *Objekte*, die dem Konzept zwecks formaler Interpretation zugeordnet werden. Die einzelnen Objekte, die in der Extension eines Konzepts enthalten sind, werden als dessen *Instanzen* bezeichnet. Beide Begriffe werden in Abschnitt 3.1.3 auch formal präzisiert. Zudem wird in der vorliegenden Arbeit der Begriff der Extension in einem weiter gefassten Verständnis aufgefasst. Als Instanzen eines Konzepts werden nämlich nicht nur einzelne Objekte, sondern auch Mengen von Objekten zugelassen. Konzepte, deren Instanzen Mengen von Objekten sind, werden als *mengenwertige Konzepte* vorgestellt.

2) Vgl. MÄDCHE ET AL. (2000), S. 131. Zur Bedeutung von Konzept-Taxonomien für Ontologien vgl. JONES/PATON (1999), S. 99 ff.; GUARINO (1999), S. 222 ff. u.a. mit der Anmerkung: „*All* Ontologies are centered on a taxonomy, based on a partial ordering relation named in various ways.“ (GUARINO (1999), S. 222 [kursive Hervorhebung durch den Verfasser]).

3) Vgl. RAUH/STICKEL (1997), S. 38 ff.; VOSSEN (1994), S. 54 f.

4) Vgl. FOWLER (2004), S. 45 f.

5) Vgl. BERGAMASCHI/SARTORI (1992), S. 386. Für eine *kognitionswissenschaftliche* Perspektive auf die taxonomische Verfeinerung sprachlicher Ausdrucksmittel zu Zwecken der Komplexitätsreduktion vgl. LÖBNER (2003), S. 272 ff. Vgl. darüber hinaus KASCHEK (2004), S. 75 ff. zum reduktionistischen Charakter von Abstraktionen.

6) Vgl. FRANK (2003), S. 14.

für dessen Subkonzepte auch zulassen.[1)] Hierunter fällt insbesondere die *Vererbung* von Relationen, anhand derer Instanzen des Konzepts zueinander in Beziehung gesetzt werden können. Beispielsweise sollte die Relation *arbeitet_mit*, mit der zwei Instanzen des Konzepts *Mitarbeiter* miteinander verknüpft werden können, auch alle Instanzen der Subkonzepte *wissenschaftlicher_Mitarbeiter* und *administrativer_Mitarbeiter* aufnehmen können.

- **Inkompatibilität**

Wenn zu einem Konzept mehrere Subkonzepte angegeben werden, bleibt hiermit zunächst ungeklärt, in welcher Beziehung die Subkonzepte zueinander stehen.[2)] Von allen denkbaren metasprachlichen Relationen, die zwischen Konzepten vorliegen können, sind insbesondere die *Inkompatibilität* und die *Äquivalenz* von Konzepten von Interesse. Während mit der Inkompatibilität der Ausschluss von Objekten aus der Extension des einen Konzepts eingefordert wird, wenn es bereits in der Extension des damit inkompatiblen Konzepts enthalten ist, wird mit der Äquivalenz die Gleichheit der Extensionen zweier Konzepte eingefordert.

Mit der Anforderung nach Spezifizierbarkeit von *Inkompatibilitäten* kann dazu Stellung genommen werden, ob die Sprache es erlaubt, solche Konzeptpaare auszuzeichnen, bei denen die Extensionen der beiden Konzepte sich nicht überlappen dürfen. Die Inkompatibilität von Konzepten wird zumeist implizit angenommen. Beispielsweise kann auf die Inkompatibilität der Konzepte *Mensch* und *Computer* geschlossen werden, ohne dass dies explizit ausgezeichnet werden müsste. In Fällen allerdings, in denen auf die Inkompatibilität nur mit domänenspezifischem Wissen geschlossen werden kann, empfiehlt sich ihre explizite Spezifikation.

- **Äquivalenz**

Der Inkompatibilität von Konzepten ist die *Äquivalenz* von Konzepten entgegengesetzt. Hinsichtlich dieses Kriteriums wird untersucht, ob die jeweilige Sprache Mittel zur Verfügung stellt, mit denen ausgedrückt werden kann, dass die Extensionen zweier Konzepte immer gleich sein müssen.[3)]

Obwohl Ausdrucksmöglichkeiten für die Äquivalenz von Strukturierungseinheiten von Sprachen zur konzeptionellen Modellierung vermehrt eingefordert werden,[4)] sind die bislang beispielhaft aufgeführten Äquivalenzen auf „konstruierte Domänen“ beschränkt. Beispielsweise ist die Äquivalenz der Konzepte *gleichwinkliges_Dreieck* und *gleichseitiges_ Dreieck* damit begründet, dass sie in ihren Extensionen immer die gleichen Objekte umfassen, obwohl sie jeweils einen unterschiedlichen „Sinn“ haben. Über solche

1) Vgl. STEIMANN (2000), S. 32 f.

2) Vgl. KHAN ET AL. (2004), S. 74.

3) Vgl. BAADER/NUTT (2002), S. 52.

4) Vgl. ORTNER (1997), S. 31 f.

Äquivalenzen hinaus, die sich aus einer inhaltlichen Kongruenzbeziehung der Merkmale von Konzepten ableiten, können auch Äquivalenzen aufgrund domänenspezifischer Umstände vorliegen. Wenn beispielsweise die Extensionen der beiden Konzepte *Aufgaben_Abteilung_A* und *Aufgaben_Mitarbeiter_4711* derart miteinander verknüpft sind, dass jede Aufgabe, die der Abteilung A zugeordnet wird, automatisch auch eine Aufgabe des Mitarbeiters 4711 ist, und zudem der entsprechende Mitarbeiter keine darüber hinaus gehenden Aufgaben zugewiesen bekommt, können die beiden Konzepte als miteinander äquivalent ausgezeichnet werden.

❒ Meta-Konzepte

In Ontologien sind objektsprachliche Konstrukte enthalten, deren Eigenschaften mittels metasprachlicher Ausdrucksmittel spezifiziert werden. Die Eigenschaften von Konzepten können ausgedrückt werden, indem auf Meta-Konzepte[1)] zurückgegriffen wird. Es handelt sich hierbei um (metasprachliche) Konzepte, deren Extensionen (objektsprachliche) Konzepte enthalten.[2)] Es wird dabei angenommen, dass alle Instanzen eines Meta-Konzepts gemeinsame Merkmale aufweisen, die spezifisch für das Meta-Konzept formuliert sind.

Mit der Zusammenfassung von objektsprachlichen Konzepten in den Extensionen von Meta-Konzepten können Erstgenannte *klassifiziert*[3)] werden. Dies kann für unterschiedliche Anwendungen von Interesse sein. Eine solche Klassifikation von Konzepten kommt beispielsweise der Integrierbarkeit der konstruierten Ontologien in Programmiersprachen zu Gute. Insbesondere objektorientierte Programmiersprachen verfügen über Klassenprinzipien, mit denen sich Ontologien eher vereinbaren lassen, wenn sie in einer Sprache spezifiziert wurden, die unterschiedliche Abstraktionsebenen zulässt.

---

1) Vgl. GROSSO ET AL. (1997), S. 3 f.; WELTY/FERRUCCI (1999), S. 25 ff.; STEIMANN (2000), S. 26 („Metatypen").

2) Die Beziehungen von metasprachlichen Konzepten zu objektsprachlichen Konzepten weisen Analogien zu den Beziehungen objektsprachlicher Konzepte zu ihren Instanzen auf. Mit beiden Beziehungsarten werden die „Instanz-von"-Beziehungen zwischen jeweils sprachlichen Konstrukten ausgedrückt, wodurch stets ein Wechsel der Abstraktionsebene stattfindet. Weil taxonomische Beziehungen zwischen Konzepten (und nicht zwischen Konzepten und deren Instanzen) definiert sind, ist es unzulässig, Meta-Konzepte in einer Ontologie mittels einer *taxonomischen* Beziehungsart mit objektsprachlichen Konzepten in ein Verhältnis zu setzen; vgl. GUARINO/ WELTY(2002), S. 64; WELTY/FERRUCCI (1999), S. 27 f. Hierdurch könnte z.B. fälschlicherweise darauf geschlossen werden, dass eine Instanz eines objektsprachlichen Konzepts auch eine Instanz des metasprachlichen Konzepts sei, dem das erstgenannte Konzept in einer taxonomischen Beziehung untergeordnet ist.

3) Der Begriff *Klassifikation* wird hier im Sinne einer metasprachlichen Gruppierung von objektsprachlichen Konzepten verwendet. Die übliche Verwendung der Bezeichnung *Klassifikation* ist zumeist an der objektsprachlichen Gruppierung von Objekten ausgerichtet. Dabei werden in der Regel solche Objekte, die ein spezifisches Merkmal erfüllen, von einer *Klasse* umfasst. Zur Bedeutung der objektsprachlichen Klassifikation im Rahmen der konzeptionellen Modellierung vgl. BOMAN ET AL. (1997), S. 39 ff.; FRANK/VAN LAAK (2003), S. 35 ff. Vgl. darüber hinaus PRASSE (2002), S. 388 f. zum Begriff *Metaklasse*, der analog zu dem hier eingeführten Begriff *Meta-Konzept* verwendet wird.

Darüber hinaus kann die Klassifizierbarkeit von Ontologien auch für die *Versionierung* von Ontologien von Interesse sein. Beispielsweise kann es vorkommen, dass bestimmte Konzepte in neueren Versionen einer Ontologie nicht mehr benötigt werden. Anstatt die Konzepte ganz zu löschen, kann mittels Meta-Konzepten die Aktualität von Konzepten ausgezeichnet werden. Beispielsweise könnte ein Meta-Konzept *aktuelle_Konzepte* alle objektsprachlichen Konzepte umfassen, die in der aktuellen Version der Ontologie enthalten sind. Demgegenüber könnte das Meta-Konzept *veraltete_Konzepte* alle Konzepte umfassen, die zwar in früheren Versionen der Ontologie enthalten waren, allerdings in der aktuellen nicht mehr benötigt werden.

Schließlich sind Meta-Konzepte von besonderem Interesse für die Beurteilung der *Güte* einer Ontologie. Ein ausgereiftes formales Instrumentarium zur Beurteilung der Güte einer Ontologie auf der Basis von Meta-Konzepten wurde von GUARINO/WELTY vorgestellt.[1)] Ihr Ansatz basiert darauf, solche Meta-Konzepte zu spezifizieren, deren Eigenschaften insbesondere bei der Konstruktion der Konzept-Taxonomie zu berücksichtigen sind.

❒ Konzept-Attribute

Mit der Spezifizierbarkeit von Meta-Konzepten verwandt ist die Anforderung nach der Spezifizierbarkeit von *Konzept-Attributen*. Mit dieser Anforderung werden Möglichkeiten zur Definition von Attributen[2)] und zugehörigen Attributswerten für Konzepte erfasst. Dabei wird ein Verständnis für Attribute unterstellt, das Attributswerte als Beziehungen von Konzepten untereinander ausdrückt.[3)]

Als Konzept-Attribute kommen alle metasprachlichen Relationen in Frage, mittels derer objektsprachliche Konzepte miteinander verknüpft werden können. Insofern sind die bereits angesprochenen Relationen zur Spezifikation von Taxonomien, Inkompatibilitäten oder Äquivalenzen von Konzepten auch als Konzept-Attribute zu interpretieren. Darüber hinaus unterstützt eine Ontologie-Sprache genau dann Konzept-Attribute, wenn sie die Spezifikation *beliebiger* metasprachlicher Relationen zwischen Konzepten erlaubt. Von besonderem Interesse könnten beispielsweise solche Konzept-Attribute sein, mit

---

1) Vgl. GUARINO/WELTY (2004), S. 151 ff.; GUARINO/WELTY (2002), S. 61 ff.; GUARINO/WELTY (2000), S. 99 ff.; WELTY/GUARINO (2001), S. 53 ff.; TAMMA/BENCH-CAPON (2002), S. 43 f. u. 52 ff.

2) Es wurde bereits in Fn. 2 auf S. 40 darauf hingewiesen, dass der systemtheoretische Rahmen der vorliegenden Arbeit auch Attribute erfasst. Attribute werden dort als Relationen zwischen den Elementen eines Systems eingeführt. Ein Attributswert lässt sich entsprechend hierzu als Beziehung eines Elements zu einem anderen Element ausdrücken. Attributswerte für Konzepte lassen sich somit als Beziehungen der Konzepte zu anderen Konzepten formulieren.

3) Formal lässt sich das Ausdrücken einer Eigenschaft durch ein Konzept-Attribut als zweistellige Relation definieren. In ihrem ersten Argument nimmt eine solche Relation Konzepte auf, deren Beziehungen zu Konzepten an der zweiten Argumentstelle ausgedrückt werden. Im Gegensatz hierzu werden mit Meta-Konzepten keine Beziehungen zwischen Konzepten ausgedrückt. Die Eigenschaft, die es auszudrücken gilt, ist in einem Meta-Konzept bereits „enthalten“.

denen in der Ontologie Informationen dazu abgelegt werden können, *wann* und *von wem* ein Konzept in die Ontologie aufgenommen wurde. Dadurch würde eine zu Meta-Konzepten komplementäre Vorgehensweise der Versionierung von Ontologien unterstützt werden.

❒ Mengenwertige Konzepte

Eine Konzeptualisierung umfasst Konzepte, Relationen zwischen Konzepten und Regeln, die für die Extensionen der Konzepte und Relationen zu gelten haben. Dabei können die Extensionen von Konzepten entweder *skalare* Objekte oder *Mengen* von Objekten als Instanzen umfassen. Als *mengenwertige Konzepte* werden solche Konzepte verstanden, deren Instanzen Mengen von Objekten sind. Es kann sich hierbei um die leere Menge „$\varnothing$“, um eine einelementige Menge $\{x\}$ oder auch um eine Menge $\{x_1,...,x_n\}$ im „herkömmlichen Sinn“ handeln. Beispielsweise würde das einwertige Konzept *Mensch* in seiner Extension einzelne Menschen aufweisen. Das mengenwertige Konzept *Menschengruppe* würde hingegen in seiner Extension Mengen von Menschen aufweisen.

Mengenwertige Konzepte können dann von Interesse sein, wenn Mengen von Objekten nicht nur in ihrer *reifizierten* Form zugelassen werden sollen.[1)] Eine solche „Verdinglichung“ von Sachverhalten läuft der natürlichen Konzeptualisierung zuwider. In der Regel sind Reifikationen dann notwendig, wenn die zugrunde gelegte Ontologie-Sprache nicht genügend Ausdrucksmöglichkeiten zur Verfügung stellt, um Mengen von Objekten als Instanzen eines Konzepts zuzulassen.

❒ Konzept-Bezeichnungen

Bei Konzepten aus der Ontologie handelt es sich um sprachliche Konstrukte, mit denen Akteure ihre Wahrnehmungen strukturieren können. Konzepte sind somit stets an *eine* bestimmte Sprache gebunden. Wünschenswert ist allerdings, dass eine Ontologie in mehreren Sprachen wiederverwendet werden kann. Die Wiederverwendbarkeit von Ontologien in mehreren Sprachen kann dadurch begünstigt werden, dass *Konzept-Bezeichnungen* zugelassen werden. Es handelt sich hierbei um metasprachliche Zeichenketten, die Konzepten als ihre natürlichsprachlichen Bezeichner zugeordnet werden

---

1) Diskrete Mengen von Objekten können in ihrer reifizierten Form ausgedrückt werden, indem für jede Menge ein Objekt spezifiziert wird, das mit einer objektsprachlichen Beziehungsart (z.B. „hat_Element“) mit allen Elementen der Menge verbunden wird. Beispielsweise kann eine Menge $X=\{x_1,...,x_n\}$ durch den Ausdruck Menge(X) und n Beziehungen der Art hat_Element(X,x_1),...,hat_Element(X,x_n) ausgedrückt werden. Beispielhaft wird eine solche Reifikation mit Hilfe der Ontologie-Sprache F-Logic in Abschnitt 5.1.1.3 vorgestellt.

können. Darüber hinaus können mit den natürlichsprachlichen Bezeichnern synonyme und homonyme[1)] Zeichenketten aufgedeckt werden.

Wenn eine Ontologie-Sprache solche metasprachlichen Bezeichnungen für Konzepte zulässt, sollte sie auch Auszeichnungsmöglichkeiten für die Zeichenketten vorsehen, damit klar wird, in *welcher* Sprache die jeweilige Zeichenkette ein Bezeichner von einem Konzept ist. Das heißt, dass bei der Zuordnung einer Zeichenkette als metasprachlicher Bezeichner eines Konzepts mit angegeben werden kann, in welcher natürlichen Sprache der Bezeichner gültig ist. Analog zu der Aufdeckung von Synonymien und Homonymien in einer Sprache kann hierdurch die Übersetzung eines Konzept-Bezeichners aus einer Sprache in eine andere Sprache vereinfacht werden. Derartige Ausdrucksmöglichkeiten sind für Ontologien insbesondere im Umfeld des *Natural Language Processings*[2)] von Interesse.

❒ Konzept-Definitionen

Die Anforderung bezüglich der Spezifizierbarkeit von Konzept-Definitionen ist darauf bezogen, ob es die jeweilige Ontologie-Sprache erlaubt, die informale Semantik von Konzepten natürlichsprachlich zu formulieren.[3)] Im Gegensatz zur konventionellen prädikatenlogischen Wissensrepräsentation kann den sprachlichen Ausdrucksmitteln, die bei der ontologiegestützten Wissensrepräsentation verwendet werden, über ihre *formale* Semantik hinaus auch noch eine *informale* Semantik zugeordnet werden. Die informale Semantik eines formalsprachlichen Ausdrucksmittels wird durch natürlichsprachliche Zeichenketten ausgedrückt, die ihm zugeordnet sind. Dadurch wird der „Sinn“ eines Konzepts zwar nicht formal spezifiziert, jedoch informal beschrieben.

Die formale Semantik objektsprachlicher Ausdrucksmittel entspricht der Menge formaler Konstrukte, die in der Extension des Ausdrucksmittels enthalten sind. So werden Konzepte durch die Mengen von Objekten formal interpretiert, die ihnen zugeordnet sind. Wenn die Extension eines Konzepts keine Objekte umfasst, wird das Konzept durch die leere Menge „∅“ formal interpretiert. Dadurch werden allerdings voneinander unterschiedliche Konzepte, die jeweils leere Extensionen haben, formal gleich interpretiert. Beispielsweise würden die beiden Konzepte *Einhorn* und *deutsche_Bundespräsidentin* formal gleich interpretiert werden.

---

1) Synonymie entspricht im allgemeinen Fall dem Umstand, in dem einer Entität mehrere Zeichenketten zugeordnet sind. Homonymie entspricht dem umgekehrten Fall, in dem mehreren Entitäten eine Zeichenkette zugeordnet ist. Bei den Entitäten, denen in einer Ontologie Zeichenketten zugeordnet werden, handelt es sich um Konzepte. Konzepte selbst sind (formal-)sprachliche Konstrukte. Insofern werden in Ontologien (formal-)sprachlichen Konstrukten (natürlich-)sprachliche Konstrukte als Bezeichner zugeordnet. Auf diesen Sachverhalt wird in Abschnitt 3.1.3.2.2.3 näher eingegangen.

2) Zum Natural Language Processing vgl. CARSTENSEN ET AL. (2001), S. 10 ff.

3) Vgl. GRUBER (1993), S. 199 („In such an ontology, definitions associate the names of entities [...] with human-readable text describing what the names are meant to denote ...“).

Dadurch, dass in Ontologien neben der Spezifikation der formalen auch die Spezifikation der informalen Semantik grundsätzlich möglich ist, können Konzepte, deren Extensionen gleich sind, voneinander unterschieden werden. Darüber hinaus ist eine solche informale Semantik an menschliche Benutzer ontologiegestützter Modelle gerichtet, die die Bedeutung von Konzepten auch anhand ihrer natürlichsprachlichen Definitionen erfassen können.

Für die Zuordnung natürlichsprachlicher Zeichenketten als Definitionen von Konzepten gelten analog die Ausführungen bezüglich der Zuordnung natürlichsprachlicher Zeichenketten als Bezeichnungen. Eine Ontologie-Sprache sollte demnach die sprachspezifische Zuordnung von Konzept-Definitionen ermöglichen. Damit können für ein Konzept mehrere Zeichenketten als Definitionen des Konzepts in unterschiedlichen natürlichen Sprachen angegeben werden.

❒ Typisierung von Relationen

Bezüglich dieser Anforderung wird untersucht, ob eine Ontologie-Sprache es erlaubt, Relationen zu *typisieren*. Als Typisierung einer Relation wird die Auszeichnung ihrer Argumentstellen mit Konzepten bezeichnet. Eine Relation ist in einer Ontologie-Sprache dann typisierbar, wenn für jede ihrer Argumentstellen angegeben werden kann, zu welchen Konzepten Objekte gehören müssen, damit zwischen ihnen eine Beziehung entsprechend der Relation vorliegen kann.[1)]

Die Typisierung von Relationen entspricht metasprachlichen Integritätsbedingungen. Durch die Typisierung einer Relation wird nämlich festegelegt, ob eine Zeichenkette als „sinnvoller“ Ausdruck zugelassen ist. Beziehungen zwischen Objekten dürfen demnach nur spezifiziert werden, wenn die entsprechende Relation über den jeweiligen Objektmengen definiert ist. Als Ausdrücke werden solche Zeichenketten ausgeschlossen, die nicht aus der typgerechten Anwendung einer Relation hervorgehen.

1) Somit wird für die Typisierbarkeit von Relationen die Zurechenbarkeit von Objekten zu Konzepten vorausgesetzt. Damit eine Beziehung zwischen Objekten entsprechend einer Relation zulässig ist, müssen die Objekte jeweils Instanzen derjenigen Konzepte sein, mit denen die Relation an entsprechender Stelle typisiert ist.

❐ Relationen-Ordnung

Mit dem zweiten Kriterium *Ordnung von Relationen* kann dazu Stellung genommen werden, ob es eine Ontologie-Sprache erlaubt, Unter- und Überordnungsbeziehungen zwischen Relationen auszudrücken.[1)] Dabei ist eine erste Relation einer zweiten Relation genau dann untergeordnet, wenn eine Beziehung zwischen Objekten aus der ersten Relation auch immer eine Beziehung zwischen den gleichen Objekten aus der zweiten Relation ist.

Die Spezifikation von Relations-Ordnungen ist ein Baustein der ontologiegestützten Modellierung für die Erschließung des „Sinns" von sprachlichen Ausdrucksmitteln. Dabei besteht eine Analogie zu der Spezifikation von Konzept-Taxonomien. Bei Letztgenanntem steht es im Vordergrund, auf die Zugehörigkeit eines Objekts zu der Extension eines Konzepts zu schließen, wenn es bereits in der Extension eines Subkonzepts davon enthalten ist. Analog kann auf die Beziehung zwischen Objekten entsprechend einer Relation geschlossen werden, wenn die Relation einer weiteren Relation übergeordnet ist und entsprechend dieser zweiten Relation eine Beziehung zwischen den Objekten spezifiziert ist.

❐ n-stellige Relationen

Eine Ontologie-Sprache sollte es ermöglichen, auch solche Relationen zu spezifizieren, mittels derer Beziehungen zwischen mehr als zwei Objekten ausgedrückt werden können. Ist die Anforderung nicht erfüllt, so erlaubt die Sprache in der Regel nur die Spezifikation von zwei-stelligen Relationen und somit auch nur Ausdrücke mit Beziehungen zwischen genau zwei Objekten.

Relationen, deren Instanzen n-Tupel von Objekten mit n>2 sind, werden für die möglichst natürliche Spezifikation einer Konzeptualisierung benötigt. Oftmals reichen zwar zwei-stellige Relationen aus, um alle Beziehungen ausdrücken zu können. Allerdings werden hierfür in der Regel „formale Krücken", wie z.B. die bereits erwähnte *Reifikation*, benötigt. Bei der Reifikation n-stelliger Relationen wird für die auszudrückende Relation ein Konzept spezifiziert, das mittels n Relationen mit den Konzepten verbunden wird, mit denen die n-stellige Relation in ihrer natürlichen Repräsentation spezifiziert würde.[2)] Instanzen dieses Konzepts entsprechen dann Instanzen der n-stelligen Relation. Solche Konstruktionen widerstreben allerdings dem Vorhaben, mit Ontologien eine möglichst *natürliche* Rekonstruktion von Begriffswelten zu vollziehen. Daher wird an Ontologie-Sprachen die Anforderung gestellt, auch solche Relationen ausdrücken zu können, deren Instanzen mehr als zwei Objekte miteinander verbinden können.

❐ Relations-Attribute

---

1) Vgl. GROSSO ET AL. (1997), S. 3; MÄDCHE ET AL. (2000), S. 135.

2) Die Reifikation n-stelliger Relationen mit n>2 wird in Abschnitt 5.1.1.3 anhand der Ontologie-Sprache F-Logic beispielhaft verdeutlicht.

Relations-Attribute können dazu verwendet werden, Eigenschaften objektsprachlicher Relationen auszudrücken. Insofern handelt es sich bei Relations-Attributen um metasprachliche Ausdrücke. Sie können einerseits dazu verwendet werden, metasprachliche Beziehungen zwischen objektsprachlichen Relationen auszudrücken. Beispielsweise kann die Unterordnung einer Relation gegenüber einer weiteren Relation in Form eines Relations-Attributs ausgedrückt werden.

Über Beziehungen zwischen Relationen hinaus können durch Relations-Attribute auch Eigenschaften von Relationen ausgedrückt werden, anhand derer Schlussfolgerungen durchgeführt werden können. Von besonderem Interesse sind hierbei Eigenschaften, die von Ordnungseigenschaften erfüllt werden. Als wesentliche Ordnungseigenschaft von Relationen kommt beispielsweise die *Transitivität*[1)] in Frage.

❐ Funktionale Relationen

Eine funktionale Relation liegt genau dann vor, wenn ein Objekt mit höchstens einem weiteren Objekt in der entsprechenden Beziehung stehen darf. Eine Sprache erfüllt die Anforderung *funktionale Relationen* genau dann, wenn sie es erlaubt, Relationen auszuzeichnen, die für funktionale Beziehungen stehen.

Funktionale Relationen sind Ausdrucksmittel, in denen implizit Anforderungen an die Integrität ontologiegestützter Modelle enthalten sind. Die Integrität eines ontologiegestützten Modells ist nämlich dann verletzt, wenn mindestens ein Objekt existiert, das in mindestens zwei Beziehungen – entsprechend der funktionalen Relation – zu zwei unterschiedlichen Objekten steht. Instanzen funktionaler Relationen verknüpfen nämlich Objekte stets mit höchstens einem weiteren Objekt. Beispielsweise wäre die Integrität einer Modells verletzt, wenn zu einer funktionalen Relation *hat_Seriennummer*, anhand derer die Seriennummern von Produkten spezifiziert werden können, zwei Instanzen existieren würden, denen zufolge ein Produkt zwei unterschiedliche Seriennummern hätte.

❐ Kardinalitäten

Mit Kardinalitäten zu Relationen werden ganzzahlige Unter- oder Obergrenzen für das Vorkommen von Objekten aus den Argumentstellen der Relation festgelegt. Entsprechend wird in eine *Minimum*- und eine *Maximumkardinalität* unterschieden.

Für eine n-stellige Relation ist die jeweilige Kardinalität immer auf eine bestimmte Stelle bezogen. Die n-te Minimumkardinalität zu einer Relation gibt an, in wie vielen entsprechenden Beziehungen ein Objekt an der entsprechenden Stelle der Relation mindestens vorkommen muss. Mit der n-ten Maximumkardinalität wird umgekehrt angegeben, in wie vielen entsprechenden Beziehungen ein Objekt an der entsprechenden Stelle der Relation höchstens vorkommen darf.

---

1) Transitive Relationen kommen in der vorliegenden Arbeit oftmals zur Geltung. Eine beispielhafte Spezifikation transitiver Relationen erfolgt in Abschnitt 5.1.1.3.

Beispielsweise können für eine Relation *kooperiert_mit* an der ersten Argumentstelle (Unternehmen) die Minimumkardinalität 0 und die Maximumkardinalität n angeben werden. Einerseits können nämlich Unternehmen existieren, die in keiner Kooperationsbeziehung zu anderen Unternehmen stehen. Anderseits können Unternehmen beliebig viele Kooperationsbeziehungen eingehen. Bei einem zweiten Beispiel *arbeitet_fuer*, das an der ersten Argumentstelle mit dem Konzept *Mitarbeiter* und an der zweiten Argumentstelle mit dem Konzept *Unternehmen* typisiert ist, würden für die erste Argumentstelle sowohl für die Minimum- als auch für die Maximumkardinalität 1 angegeben werden. In diesem Fall muss jeder Mitarbeiter zu genau einem Unternehmen in einer *arbeitetfuer*-Beziehung stehen. Für die zweite Argumentstelle würden als Minimumkardinalität 1 und als Maximumkardinalität n angegeben werden, da Unternehmen mindestens einen und höchstens beliebig viele Mitarbeiter haben müssen.

❐ Relationen-Bezeichner

Für Relationen gelten die Ausführungen bezüglicher natürlichsprachlicher Bezeichner für Konzepte analog. Demnach sollte es eine Ontologie-Sprache erlauben, einer Relation mehrere natürlichsprachliche Bezeichner zuzuordnen. Auch hierbei gilt, dass die Ontologie-Sprache Ausdrucksmittel zur Verfügung stellen sollte, anhand derer die Sprachspezifität der jeweiligen Bezeichner angegeben werden kann.

❐ Relationen-Definition

Wie schon für Konzepte eingefordert, sollte eine Ontologie-Sprache es ermöglichen, natürlichsprachliche Definitionen für Relationen anzugeben, wodurch der „Sinn" von Relationen informal beschrieben werden kann. Auch hierbei gilt, dass Relations-Definitionen sprachspezifisch angebbar sein sollten.

Von dem dritten Teilkatalog für die statische Modellierungsfähigkeit werden Kriterien erfasst, mittels derer die Spezifizierbarkeit *regelartiger Zusammenhänge* beurteilt werden kann. Hierunter fallen sowohl Regeln, mittels derer die Zulässigkeit von Aussagen überprüft werden kann, mit denen ein Akteur sein Wissen ontologiegestützt ausdrückt, als auch solche Regeln, mittels derer sich aus dem vorhandenen Wissen des Akteurs neues Wissen erschließen lässt. Entsprechend dieser Zweiteilung von Regeln werden im Anforderungskatalog die beiden Anforderungen *Integritätsregeln* und *Inferenzregeln* verwendet. Als Integritätsregeln werden solche Regeln behandelt, die zu Zwecken der Zulässigkeitsüberprüfung von Aussagensystemen konstruiert werden. Inferenzregeln werden dazu konstruiert, aus vorhandenen Aussagen auf neue Aussagen zu schließen.

Die Spezifikation regelartiger Zusammenhänge wird insbesondere unter dem Schlagwort „Business-Rules" seit einigen Jahren sowohl in der Informatik als auch in der Wirtschaftsinformatik intensiv diskutiert.[1)] Ausgehend von dem Leistungspotenzial von

---

1) Vgl. ERWIN (2002), S. 143 ff.; FRANK/VAN LAAK (2003), S. 83; KNOLMAYER ET AL. (2000), S. 18 ff. OBERWEIS (1996), S. 197; RAM/KHATRI (2005), S. 89 ff.; ROSCA ET AL. (2002), S. 363 ff.

Inferenz- und Integritätsregeln ist gar die „neue“ Forschungsrichtung *Business Rule-Oriented Conceptual Modeling* aufgekommen,[1] die sich allerdings in erster Linie an den Prinzipien orientiert, die bereits für deduktive Datenbanksysteme seit längerem bekannt[2] sind. Um einerseits die Anschlussfähigkeit an diese bestehenden Forschungsrichtungen nicht zu verlieren, sind Inferenz- und Integritätsregeln für Ontologien ein wesentlicher Faktor. Andererseits wird die ontologiegestützte Modellierung durch Ausdrucksmittel für Regeln gegenüber traditionellen Modellierungssprachen eindeutig profiliert.[3] Daher werden an Ontologie-Sprachen die Anforderungen nach der Spezifizierbarkeit von Integritäts- und Inferenzregeln gestellt.

❐ Integritätsregeln

Eine Ontologie-Sprache verfügt genau dann über die Mächtigkeit, Integritätsregeln auszudrücken, wenn sie es erlaubt, solche Ausdrücke zu konstruieren, mit denen eine zulässige Wissensbasis nicht im Widerspruch sein darf. Stehen Aussagen aus einer ontologiegestützten Wissensbasis im Widerspruch zu einer Integritätsregel, so ist die Wissensbasis bezüglich der Integritätsregel unzulässig. Sie ist wiederum genau dann bezüglich einer Integritätsregel zulässig, wenn keine Aussagen aus der Wissensbasis im Widerspruch zu der Integritätsregel stehen.

Beispielsweise könnte eine Integritätsregel spezifiziert werden, der zufolge Produkte einer bestimmten Sorte aufgrund physikalischer Umstände bestimmte Maße nicht überschreiten dürfen. Eine ontologiegestützte Wissensbasis, in der mindestens eine Aussage enthalten ist, der zufolge die Maße eines Produktes der entsprechenden Sorte die durch die Integritätsregel als Grenze festgelegte Größe überschreiten, ist unzulässig.

---

1) Vgl. HERBST (1997).

2) Vgl. BOMAN ET AL. (1997), S. 64 ff.; KUMAR DAS (1992), S. 275 ff.; RAUH/STICKEL (1997), S. 69 ff.; VOSSEN (1994), S. 510 ff.

3) Beispiele für etablierte Modellierungssprachen, die weder Inferenz- noch Integritätsregeln unterstützen, sind die UML und Entity-Relationship-Diagramme. Die Profilierung formaler Sprachen gegenüber den Entity-Relationship-Diagrammen umschreibt SOWA auf plakative Weise wie folgt:

> „A major problem with E/R diagrams and similar formalisms is that they are strictly less powerful than formal logic. As a consequence, they are incapable of dealing with certain important design features – in particular, anything involving quantifiers, which includes most integrity constraints – that formal logic can handle. (The quantifiers were invented by Frege in 1879, which makes E/R diagrams a pre-1879 kind of logic!)“ – zitiert nach ERDMANN (2001), S. 132.

❐ Inferenzregeln

Inferenzregeln dienen dazu, Wissen, dass in den explizit repräsentierten Fakten einer Wissensbasis implizit enthalten ist, zu erschließen. Die Ontologie-Sprache verfügt genau dann über die Mächtigkeit, Inferenzregeln auszudrücken, wenn sie es erlaubt, Ausdrücke zu formulieren, anhand derer aufgrund vorliegender Fakten neue Fakten erschlossen werden können.

Mit Inferenzregeln werden Ontologie-Sprachen tendenziell[1)] in ihrer „Ausdrucksökonomie" angereichert. Während in konventionellen Modellierungssprachen ohne Ausdrucksmöglichkeiten für Inferenzregeln keine Unterscheidung zwischen expliziten und impliziten Fakten getroffen wird, da die „Implizitheit" eines Faktums stets in Bezug auf eine Inferenzregel und ein explizites Faktum definiert ist, ist dies für Ontologie-Sprachen grundsätzlich möglich. Die Ausdrucksökonomie von Ontologie-Sprachen wird dadurch erhöht, dass bei der Verfügbarkeit von Inferenzregeln tendenziell auf Aussagen geschlossen werden kann, die unter sonstigen Umständen auf ressourcenverzehrende Weise explizit spezifiziert werden müssten.

❐ Regel-Definitionen

Integritäts- und Inferenzregeln werden im Allgemeinen entsprechend Regeln zur Konstruktion formaler Ausdrücke konstruiert, denen zufolge Ausdrücke beliebiger Komplexität zugelassen sein können. Demnach können Inferenz- und Integritätsregeln eine Komplexität aufweisen, die es menschlichen Akteuren erschweren kann, die Regeln zu verstehen. Regel-Definitionen können es unter solchen Umständen menschlichen Ontologie-Nutzern erleichtern, die Bedeutung von Inferenz- und Integritätsregeln nachzuvollziehen.

Analog zu den entsprechenden Kriterien aus dem ersten und dem zweiten Abschnitt des Anforderungskatalogs für die statische Modellierungsfähigkeit können mit dem Kriterium *Regel-Definitionen* dazu Aussagen gemacht werden, ob es eine Ontologie-Sprache erlaubt, Integritäts- und Inferenzregeln natürlichsprachliche Definitionen zuzuordnen. Dabei ist das Kriterium *Regel-Definitionen* den Kriterien *Integritätsregeln* und *Inferenzregeln* insofern untergeordnet, als dass Definitionsmöglichkeiten für Regeln nur dann gefordert werden können, wenn die entsprechende Sprache auch die Spezifikation von

---

1) Bezüglich der „Ausdrucksökonomie" von Inferenzregeln lässt sich aufgrund zweier Faktoren lediglich eine Tendenzaussage treffen. Zum einen kann auf die explizite Spezifikation von Fakten verzichtet werden, wenn sie sich aus bereits explizit spezifizierten Fakten mit Hilfe von Inferenzregeln erschließen lassen. Insofern weisen Inferenzregeln ein *Ressourceneinsparpotenzial* auf. Es können jene Ressourcen eingespart werden, die bei „redundanter" Spezifikation von Fakten Einsatz finden würden. Zum anderen bedarf die Spezifikation von Inferenzregeln selbst des Einsatzes von Ressourcen. Darüber hinaus kann auch die Anwendung von Inferenzregeln auf explizite Fakten zwecks Erschließung impliziter Fakten Ressourcen verzehren. Nur wenn der Ressourcenaufwand, der durch die Spezifikation und Anwendung von Inferenzregeln entsteht, geringer als die Ressourceneinsparung verbleibt, die durch den Verzicht auf die Spezifikation expliziter Fakten entsteht, kann von einer „ökonomischeren" Ausdrucksvariante gesprochen werden.

Inferenz- oder Integritätsregeln unterstützt. Wird die Spezifikation von Regeln von einer Ontologie-Sprache nicht unterstützt, entfällt auch die Anforderung der Spezifikation von Definitionen für Regeln.

❒ **Regel-Bezeichner**

Regel-Bezeichner haben mit Regel-Definitionen gemeinsam, dass es sich jeweils um metasprachliche Ausdrücke handelt, die objektsprachlichen Konstrukten zugeordnet werden. Im Gegensatz zu Regel-Definitionen dienen jedoch Regel-Bezeichner nicht der natürlichsprachlichen Erläuterung des Inhalts von Regeln. Regel-Bezeichner fungieren als natürlichsprachliche „Identifikatoren" von Regeln. Durch ihre Bezeichnung wird einer Regel eine Zeichenkette zugeordnet, die es menschlichen Benutzern vereinfacht, die Regeln zu identifizieren.

2.1.3.2.3 Anforderungskatalog für die dynamische Modellierungsfähigkeit

Der Anforderungskatalog für die *dynamische Modellierungsfähigkeit* umfasst solche Anforderungen, die für die Modellierung der dynamischen Aspekte eines Realitätsausschnitts von Bedeutung sind. Analog zum Anforderungskatalog für die statische Modellierungsfähigkeit kann mit diesem zweiten Anforderungskatalog die dynamische Modellierungsfähigkeit des integrativen Modellierungskonzepts beurteilt werden.

❒ **Einfache Prozesse**

Das erste Kriterium zur Bestimmung der dynamischen Modellierungsfähigkeit ist die Möglichkeit, *einfache Prozesse* zu spezifizieren.[1)] Als Prozesse werden hierbei Folgen aus streng alternierenden *Zuständen* und *Ereignissen* verstanden,[2)] wobei angenommen wird, dass jeder Prozess über einen Start- und einen Schlusszustand verfügt. Während Zustände zeitverbrauchender Natur sind und keine Veränderung der systembeschreibenden Größen bewirken, sind Ereignisse zeitlos. Im Gegensatz zu Zuständen bewirken Ereignisse eine Änderung von systembeschreibenden Größen. Es wird sowohl für die Zustands- als auch für die Ereignismenge eines einfachen Prozesses vorausgesetzt, dass sie nicht-leer sind. Darüber hinaus wird festgelegt, dass für jedes Ereignis ein (Folge-)Zustand definiert ist. Insofern wird als kleinster möglicher Prozess die Verknüpfung zweier Zustände mit einem Ereignis zugelassen.

Einfache Prozesse sind solche Prozesse, bei denen eine *sequentielle* Anordnung von Ereignissen vorliegt. Die sequentielle Anordnung von Ereignissen kann in Form einer totalen Ordnung auf der Menge aller prozesszugehörigen Ereignisse präzisiert werden. Im Fall eines einfachen Prozesses kann für jedes Ereignis e_1 durch eine Ordnungsrelation

---

1) Zur Bedeutung der *Spezifizierbarkeit einfacher* oder *sequentieller* Prozesse vgl. FRANK/VAN LAAK (2003), S. 77; VAN DER AALST/VAN HEE (2002), S. 4; ZELEWSKI (1996), S. 370.

2) Vgl. ZELEWSKI (1995), Bd. 2, S. 101.

bestimmt werden, ob es einem zweiten Ereignis e_2 gegenüber kausal vor- oder nachgelagert ist. Somit kann in einem einfachen Prozess kein Ereignispaar e_1 und e_2 vorliegen, das weder in der Form (e_1,e_2) noch in der Form (e_2,e_1) in der Ordnungsrelation für Ereignisse enthalten ist. Insofern konstituiert die Ordnungsrelation, durch die die kausale Anordnung von Ereignissen in einem einfachen Prozess ausgedrückt wird, eine totale Ordnung.

Durch Ereignisse können operationale Variationen eines ontologiegestützten Modells bewirkt werden. Entsprechend sollte die Erweiterung von Ontologien um dynamische Aspekte solche Prozesse repräsentieren können, bei denen Variationen eines ontologiegestützten Modells aufgrund nacheinander folgender Ereignisse stattfinden können. Dabei lassen sich die operationalen Variationsmöglichkeiten für ontologiegestützte Modelle auf *zustandsändernde Grundoperationen*[1] zurückführen. Durch diese Grundoperationen werden zustandsändernde Ereignisse konzeptualisiert. Dieser Konzeptualisierung entspricht die Einschränkung von Grundoperationen auf Fakten. Letztgenannte wurden als konstitutive Bausteine ontologiegestützter Modelle vorgestellt.

Als Grundoperationen kommen *Einfüge-*, *Lösch-* und *Änderungsoperationen* in Frage. Einfügeoperationen sind solche Ereignisse, bei denen das ontologiegestützte Modell um ein weiteres Faktum angereichert wird. Dem entgegengesetzten Fall entsprechen Löschoperationen, bei denen Fakten aus dem Modell entfernt werden. Eine Kombination aus Einfüge- und Löschoperationen sind Änderungsoperationen. Ereignisse, die sich als Änderungsoperationen charakterisieren lassen, weisen sowohl Aspekte der Faktenentfernung als auch Aspekte des Fakteneinfügens auf.

❒ Komplexe Prozesse

Die Spezifizierbarkeit einfacher Prozesse ist ein Grenzfall der Spezifizierbarkeit *komplexer Prozesse*. Unter komplexe Prozesse fallen solche Prozesse, die aus mindestens zwei Teilprozessen bestehen. Jeder Teilprozess aus einem komplexen Prozess kann entweder komplex sein oder einen einzelnen einfachen Prozess darstellen. Für komplexe Prozesse gilt, dass zwischen den darin enthaltenen verschiedenen Teilprozessen interprozessuale Abhängigkeiten existieren können, aber nicht müssen.[2] Von der ersten Variante komplexer Prozesse werden auch einfache Prozesse erfasst.

Die Inklusion einfacher Prozesse durch das Konzept komplexer Prozesse findet sich in der Ordnungsrelation wieder, durch die die kausale Anordnung von Ereignissen reprä-

---

1) Der Begriff *Operation* wird in der vorliegenden Arbeit insofern homonym verwendet, als dass er zum einen für Aktivitäten steht, durch die Zustandsübergänge in einem System bewirkt werden. Zu einem solchen Verständnis für den Begriff Operation vgl. MORDAU (1998), S. 93 ff. und 129 ff. Zum anderen wird der Begriff auch für rechtseindeutige Relationen verwendet. Auf diese Verwendung wurde bereits im systemtheoretischen Rahmen (vgl. Abschnitt 2.1.2) hingewiesen. Verwechslungen zwischen den jeweiligen Begriffsverwendungen sind nicht zu befürchten, da stets aus dem Argumentationskontext hervorgeht, von welchem Verständnis ausgegangen wird.

2) Vgl. KOUBARAKIS/PLEXOUSAKIS (2002), S. 304.

sentiert wird. Im Gegensatz zu einfachen Prozessen muss durch die Ordnungsrelation keine totale Ordnung konstituiert werden. Insofern kann für zwei voneinander unterschiedliche Ereignisse e_1 und e_2 in einem komplexen Prozess gelten, dass weder (e_1,e_2) noch (e_2,e_1) in der Ordnungsrelation enthalten sind. Wenn für alle Paare von Ereignissen e_1 und e_2 entweder (e_1,e_2) oder (e_2,e_1) in der Ordnungsrelation enthalten ist, handelt es sich um einen einfachen Prozess.

Die Spezifizierbarkeit komplexer Prozesse steigt mit den Möglichkeiten zur Bestimmung von Interdependenzen zwischen den Teilprozessen. Solche Interdependenzen können z.B. vorliegen, wenn Teilprozesse miteinander in einem *konkurrenten* Verhältnis stehen. Das konkurrente Verhältnis zwischen Teilprozessen kann zum Beispiel im Fall von knappen Ressourcen vorliegen, die von beiden Teilprozessen beansprucht werden und nicht für die gemeinsame Durchführung beider Teilprozesse ausreichen. Den dazu entgegengesetzten Fall umfasst die *Nebenläufigkeit* von Teilprozessen. Sie liegt genau dann vor, wenn die betrachteten Teilprozesse voneinander unabhängig durchgeführt werden können.

An das Modellierungskonzept wird die Anforderung nach Mitteln zum Ausdrücken von komplexen Prozessen gestellt. Entsprechend den Ausführungen zu komplexen Prozessen umfasst die Anforderung Ausdrucksmöglichkeiten für gegenseitige Abhängigkeiten oder Unabhängigkeiten zwischen Teilprozessen. Dieser Aspekt hat eine hohe Bedeutung für die Modellierung kooperativer Informationssysteme, da die Teilprozesse der beteiligten Akteure als Module eines Gesamtprozesses spezifizierbar sein müssen.

❒ Entscheidungsalternativen

In bestimmten Zuständen kann es vorkommen, dass Bedingungen erfüllt sind, aufgrund derer unterschiedliche Ereignisse stattfinden können. Unter Umständen kann es dabei sogar sein, dass die unterschiedlichen Ereignisse sich wechselseitig ausschließen. Ein solcher wechselseitiger Ausschluss liegt z.B. im Fall der bereits erwähnten Knappheit von solchen Ressourcen vor, die beim Eintreten von unterschiedlichen Ereignissen beansprucht werden. In einer solchen Konkurrenzsituation können nicht alle Ereignisse gleichzeitig durchgeführt werden, wenn das Ressourcenangebot nicht den gemeinsamen Bedarf abdeckt.

Prozesse, in denen mindestens zwei Ereignisse alternativ verknüpft vorkommen, werden als *nichtdeterministische Prozesse* bezeichnet. Nichtdeterministische Prozesse zeichnen sich dadurch aus, dass a priori nicht festgelegt werden kann, welcher konkrete Ablauf stattfinden muss, da in dem Prozess *Entscheidungsalternativen*[1)] vorhanden sind, bezüglich derer keine modellendogene Entscheidung getroffen wird. In diesen Fällen muss eine *modellexogene* Entscheidung getroffen werden, welche der beiden Ereignisse stattfinden soll. Die Gründe, die für die Durchführung eines der zueinander alternativen Er-

1) Vgl. ZELEWSKI (1996), S. 370.; FRANK/VAN LAAK (2003), S. 78 f.

eignisse sprechen, werden hierbei nicht im Modell repräsentiert, sondern müssen von dem Modellierungsträger bestimmt werden.[1)]

An das Modellierungskonzept wird die Anforderung gestellt, auch die Spezifikation von nicht-deterministischen Prozessen zu unterstützen. Zu diesem Zweck sollte das Modellierungskonzept in der Lage sein, Entscheidungsalternativen zu berücksichtigen. Hierzu gehört, dass in jedem Zustand eines ontologiegestützten Modells kenntlich gemacht werden kann, welche Ereignisse eintreten können. Ist das alternative Eintreten mehrerer voneinander unterschiedlicher Ereignisse möglich, sollte das Modellierungskonzept Ausdrucksmittel für die ereigniseintrittsbezogenen Entscheidungsalternativen zur Verfügung stellen.

❒ Synchronisation

Wenn das Modellierungskonzept komplexe Prozesse unterstützt, kann es nötig sein, für nebenläufige Teilprozesse einen oder mehrere gemeinsame *Folgeprozesse* anzugeben. Durch den gemeinsamen Folgeprozess werden die Teilprozesse wieder *synchronisiert*.[2)] Die Synchronisation von Teilprozessen ist ein Konzept der Prozesskoordinierung, das in unterschiedlichen Zuständen erforderlich sein kann.

Ausgelöst wird der Synchronisationsbedarf oftmals durch einen „Fork“. Es handelt sich hierbei um die Splittung eines komplexen Prozesses aufgrund eines Ereignisses in mindestens zwei wechselseitig unabhängige, also nebenläufige Teilprozesse. Zu Zwecken der Koordination kann es nötig sein, solche voneinander unabhängige Teilprozesse – in Form eines „Joins“ – zusammenzuführen. Um derartige Teilprozesse koordinieren zu können, wird an das Modellierungskonzept die Anforderung gestellt, Mittel für die Synchronisation von Teilprozessen zur Verfügung zu stellen.

❒ Iterative Prozesse

Wenn Prozesse mehrmals ablaufen müssen, ist ein Konzept notwendig, um die Bedingungen angeben zu können, unter denen die Prozesse entweder wiederholt werden oder aber terminieren. Bei *iterativen Prozessen* handelt es sich um Prozesse, deren mehrmaliger Ablauf erwünscht ist. Wie bei an deren Prozesstypen auch, wird vor dem Ablauf iterativer Prozesse die Erfülltheit von bestimmten Bedingungen überprüft. In der Regel geht die Erfülltheit von Bedingungen aus den zustandsspezifischen Belegungen von *Kontrollvariablen* hervor. Entsprechend diesen zustandsspezifischen Belegungen der Kontrollvariablen wird nach dem Durchlaufen des Prozesses entschieden, ob der Prozess schleifenartig erneut durchlaufen wird oder ob er terminiert. Mit der Termination

---

1) Alternativ zu einer modellexogenen Entscheidung im Fall zueinander alternativer Ereignisse kann auch eine modellendogene Bestimmung anhand *stochastischer* Verteilungsparameter, die durch einen Zufallszahlengenerator gesteuert sind, durchgeführt werden; vgl. ROSENKRANZ (2002), S. 119 ff. Modellierungskonzeptionen, die auf stochastische Parameter zurückgreifen, liegen allerdings außerhalb des Erkenntnisinteresses der vorliegenden Arbeit.

2) Vgl. FRANK/VAN LAAK (2003), S. 81 f.; VAN DER AALST/VAN HEE (2002), S. 5.

des Prozesses kann zu einem Folgeprozess übergegangen werden. Insofern korrespondieren iterative Prozesse mit den iterativen Programmierkonzepten höherer Programmiersprachen.[1)]

Das Modellierungskonzept sollte über Mittel verfügen, iterative Prozesse zu repräsentieren. Dabei sollten die Terminierungsbedingungen für die schleifenartig zu durchlaufenden Prozesse unterschiedlicher Natur sein dürfen. Als Terminierungsbedingung kommt beispielsweise eine Obergrenze für die Anzahl der Iterationen in Betracht. Der Prozess terminiert in diesem Fall erst dann, wenn die Obergrenze für die Schleifendurchläufe erreicht ist. Eine weitere Variante iterativer Prozesse liegt bei Prozessen vor, deren Termination davon abhängt, ob ein Satisfizierungsziel erreicht wird. Beispielsweise kann als Kontrollvariable ein produktspezifischer Umsatz verwendet werden. Der Produktions- und Absatzprozess für ein Produkt könnte in diesem Fall solange wiederholt werden, bis ein Umsatzziel erreicht ist. Über derartige modellexogene Kontrollvariablen hinaus sollten auch Terminierungsbedingungen für iterative Prozesse zugelassen werden, deren Erfülltsein modellendogen bestimmt wird.

❐ Pre- und Postconditions

Die Formulierbarkeit von Bedingungen, von denen das Eintreten von Ereignissen abhängen kann, wurde bereits im Kontext der Anforderung *Entscheidungsalternativen* angesprochen. Mit dem Kriterium *Pre-* und *Postconditions* wird die Anforderung nach der Formulierbarkeit von Bedingungen detailliert. Hierunter fallen einerseits Vorbedingungen (Preconditions), die erfüllt sein müssen, damit ein Ereignis stattfinden kann, und andererseits Nachbedingungen (Postconditions), die erfüllt sein müssen, nachdem das Ereignis stattgefunden hat.[2)]

Die Bezeichnungen Pre- und Postcondition sind jeweils in Bezug auf ein Ereignis definiert. Eine Bedingung kann nämlich einerseits eine Postcondition für ein erstes Ereignis sein, wenn sie erfüllt sein muss, sobald das Ereignis stattgefunden hat, und zugleich eine Precondition für ein zweites Ereignis sein, wenn sie erfüllt sein muss, damit dieses Ereignis stattfinden kann. Zudem kann eine Bedingung sowohl eine Pre- als auch eine Postcondition desselben Ereignisses sein. In diesem Fall muss eine Bedingung sowohl *vor* als auch *nach* dem Eintreten eines Ereignisses erfüllt sein.

---

1) Vgl. FRANK/VAN LAAK (2003), S. 80 f.; JENSEN (1996), S. 35 f.

2) Vgl. FRANK/VAN LAAK (2003), S. 84 f.

❐ Priorisierung deklarativer Regeln

Eine Methode, mittels derer eine Erweiterung von Ontologien um dynamische Aspekte umgesetzt werden soll, muss zwei unterschiedliche Arten von Zustandsveränderungen berücksichtigen. Beide Arten der Zustandsveränderung lassen sich auf „Wenn, ... dann ...“-Regeln zurückführen.

Die erste Art von Regeln umfasst solche Regeln, die bereits im Rahmen der statischen Struktur angesprochen wurden. Es handelt sich hierbei um Integritäts- und Inferenzregeln. Während Integritätsregeln dazu verwendet werden, unzulässige Zustände ontologiegestützter Modelle kenntlich zu machen, werden Inferenzregeln dazu benutzt, implizites Wissen zu explizieren, das in den Zuständen vorliegen kann.[1)] Integritäts- und Inferenzregeln lassen sich gemeinsam als *deklarative Regeln* ansprechen. Es handelt sich hierbei um solche Regeln, bei denen keine zeitliche Diskrepanz zwischen einerseits der Bedingung aus dem „Wenn“-Teil und andererseits der Konklusion aus dem „Dann“-Teil vorliegt. Zustände, in denen zwar der „Wenn“-, aber nicht der „dann“-Teil erfüllt ist, gelten als unzulässig. Sämtliche deklarativen Regeln aus einer Ontologie sind ein wesentlicher Baustein bei der Bestimmung der deklarativen Semantik von Ontologien. Sie werden im Rahmen der formalen Semantik ontologiegestützter Modelle stets berücksichtigt.

Die zweite Art von Regeln umfasst *Produktionsregeln*. Produktionsregeln lassen sich als Gegenstücke zu deklarativen Regeln als *prozedurale Regeln* charakterisieren. Im Gegensatz zu deklarativen Regeln kann bei prozeduralen Regeln zwischen dem „Wenn“- und dem „Dann“-Teil eine zeitliche Diskrepanz vorliegen. Sind in einem Zustand die Bedingungen für eine prozedurale Regel erfüllt, kann das Ereignis stattfinden, das in dem „Dann“-Teil der Regel enthalten ist. Tritt das entsprechende Ereignis ein, wird eine zeitlose Operation durchgeführt, die eine Zustandsvariation des ontologiegestützten Modells bewirkt.

Würde das Eintreten von Ereignissen erlaubt, obwohl die Bedingungen für mindestens eine deklarative Regel erfüllt sind, so könnte es aufgrund der Eigenart deklarativer Regeln zu folgenden Fehlern kommen. Wenn die Bedingungen für eine Integritätsregel erfüllt sind, liegt ein unzulässiger Zustand des ontologiegestützten Modells vor. In diesem Fall ist das Eintreten jeglicher Ereignisse auszuschließen, da die Preconditions der entsprechenden prozeduralen Regel zwar erfüllt aber unzulässig sein könnten. Wenn hingegen die Bedingungen für eine Inferenzregel erfüllt sind, so könnte in dem Zustand des ontologiegestützten Modells implizites Wissen enthalten sein, dessen Explikation Vorrang vor jedem Ereignis hat, durch das eine Änderung des Modellzustands bewirkt wür-

---

1) Insofern trifft der zustandsverändernde Charakter lediglich auf Inferenzregeln zu. Integritätsregeln bewirken keine unmittelbare Zustandsveränderung. Sie werden dennoch im Folgenden als zustandsverändernde Regeln thematisiert, da die Gültigkeit einer Integritätsregel in einem ontologiegestützten Modell einen zustandsverändernden Eingriff in das Modell nötig macht, um dessen Zulässigkeit wiederherzustellen, oder Zustandsveränderungen verhindert, die einen Übergang in einen unzulässigen Modellzustand darstellen würden.

de. Die Folgezustände würden allerdings in diesem Fall ihren Ursprung in einem unzulässigen Modellzustand haben.

Um unzulässige Operationen zu verbieten, muss ein Konzept zur Erweiterung von Ontologien um dynamische Aspekte eine Trennung zwischen den beiden Regelarten unterstützen. Es muss einerseits ersichtlich sein, ob ein verwendetes Konstrukt eine deklarative oder eine prozedurale Regel repräsentiert. Dies ist notwendig, um die *Priorisierung* von deklarativ bedingten Zustandsübergängen gegenüber prozedural bedingten Zustandsübergängen aufrecht zu erhalten. Erst hierdurch kann gewährleistet werden, dass die Bedingungen für prozedurale Regeln nicht in unzulässigen Modellzuständen erfüllt sein können.

❐ Konzeptendogene Objektgenerierung

Im Initialzustand eines ontologiegestützten Modells sind Objekte und Beziehungen zwischen den Objekten enthalten. Jedes Objekt ist eine Instanz mindestens eines Konzepts aus der Ontologie. Jede Beziehung ist eine Instanz einer Relation aus der Ontologie. Aufgrund von Ereignissen im Prozessablauf kann es notwendig sein, Objekte und Beziehungen zu *generieren*, die in dem Initialzustand nicht enthalten gewesen sind. Das heißt, dass die Extensionen von Konzepten und Relationen „zur Laufzeit" variiert werden. Somit können in Folgezuständen Objekte und Beziehungen enthalten sein, die in dem Initialzustand nicht enthalten gewesen sind.

Die Instanziierung von Konzepten ist notwendige Voraussetzung für die Instanziierung von Relationen. Relationen werden nämlich über den konzeptspezifischen Objektmengen definiert. Damit eine Beziehung zwischen zwei Objekten ausgedrückt werden kann, müssen beide Objekte in den entsprechenden Objektmengen bereits enthalten sein. Beispielsweise muss eine neue Produktart zunächst angelegt werden, damit der Preis für die Produktart nachfolgend als Beziehung zu einer reellen Zahl ausgedrückt werden kann.

Das Modellierungskonzept sollte die *konzeptendogene Objektgenerierung* im Prozessablauf unterstützten. Als konzeptendogene Objektgenerierung wird hierbei die Instanziierung von Konzepten und Relationen als Ergebnis des Eintretens von Ereignissen verstanden. Demnach sollte das Konzept Ausdrucksmittel zur Verfügung stellen, anhand derer Konzepte und Relationen mit Objekten bzw. Beziehungen instanziiert werden können, die im Initialzustand des Modells nicht enthalten gewesen sind.

❐ Kapazitäten

Üblicherweise werden Kapazitäten in der statischen Struktur von Ontologien nicht thematisiert. Somit wird für die verwendeten Strukturierungseinheiten implizit eine unendliche Kapazität angenommen. Dies läuft allerdings der Modellierung betrieblicher Phänomene zuwider, in der oftmals Kapazitäten anzutreffen sind. Beispielsweise kann die Anzahl an Ressourcen, die ein Akteur zeitgleich zu verarbeiten in der Lage ist, aufgrund der beschränkten Informationsverarbeitungskapazität begrenzt sein. Von einer unendlichen Kapazität der entsprechenden Strukturierungseinheit auszugehen, könnte in diesen Fällen zu unzulässigen Zustandsveränderungen führen. Im genannten Beispiel könnte es sein, dass Ereignisse zugelassen werden, aufgrund derer einem Sachbearbeiter so viele Dokumente zugeordnet werden, wie er gar nicht zu bearbeiten in der Lage ist.

Das Modellierungskonzept sollte daher Ausdrucksmittel zur Verfügung stellen, um Kapazitäten[1] von verwendeten Strukturierungseinheiten auszudrücken. Darüber hinaus sollten die Kapazitäten bei modellendogenen Zustandsvariationen berücksichtigt werden, so dass Ereignisse nicht geschehen können, bei deren Eintritt mindestens eine Kapazitätsverletzung einträfe.

❒ Zeitbezogene Determinanten

Das Eintreten von Ereignissen kann auch von zeitlichen Determinanten abhängen.[2] Beispielsweise kann es sein, dass Ereignisse – wie z.B. die monatliche Lohnabrechnung – immer zu bestimmten Zeitpunkten eintreten müssen. Genauso kann es der Fall sein, dass Prozesse eine bestimmte Zeitdauer in Anspruch nehmen. Übertragen auf die ontologiegestützte Modellierung kann ein Zustandsübergang in einem ontologiegestützten Modell u.U. aufgrund des Eintretens zeitpunktbezogener Ereignisse stattfinden. Das integrative Modellierungskonzept sollte daher in der Lage sein, zeitbezogene Determinanten zu berücksichtigen.

❒ Graphische Visualisierbarkeit

Graphische Modelle heben sich dadurch von textuellen Modelle ab, dass Erstgenannte aufgrund ihrer *Anschaulichkeit* für Benutzer tendenziell *einfacher* zu verstehen sind.[3] Dadurch wird u.a. die Verwertbarkeit eines Modells als *Kommunikationsmedium* begünstigt.[4] Einerseits wird durch die graphische Darstellung die Nachvollziehbarkeit auch für solche Benutzer ermöglicht, die mit formalen Darstellungen Schwierigkeiten hätten. Andererseits kann durch einen „Kompromiss“ zwischen natürlichsprachlicher und formaler Darstellung ein Grad an Ausdrucksmächtigkeit und Präzision erreicht werden, der für Kommunikationszwecke ausreicht.

Für die statische Struktur des integrativen Modellierungskonzepts liegen bereits Sprachen vor, anhand derer ontologiegestützte Modelle graphisch repräsentiert werden können. Hierzu gehören insbesondere *semantische Netze*. Sie werden seit längerem in unterschiedlichen Variationen diskutiert.[5] Eine besondere Aufmerksamkeit genießen hier-

---

1) Vgl. FRANK/VAN LAAK (2003), S. 66 f.

2) Vgl. BARROS/TER HOFSTEDE (1998), S. 332 ff.

3) Vgl. DUSSART ET AL. (2002), S. 8.

4) Vgl. ZELEWSKI (1995), Bd. 9, S. 71.

5) Vgl. BIBEL (1993), S. 32 ff. (mit der Bezeichnung „Assoziative Netze“); RÖBBECKE (1995), S. 47 ff.; SCHMID/ KINDSMÜLLER (1996), S. 35 ff.

bei die *Conceptual Graphs* von SOWA.[1)] Bei *Conceptual Graphs* handelt es sich um Graphen, mittels derer ein Großteil der prädikatenlogisch aufgebauten Ausdrücke rekonstruiert werden kann. Entsprechend der graphischen Darstellbarkeit der statischen Struktur sollte das Modellierungskonzept auch eine graphische Darstellung der dynamischen Struktur erlauben. Hierzu sollte der Ansatz Symbole zur Verfügung stellen, anhand derer die dynamische Struktur spezifiziert werden kann.

❒ Implementierbarkeit

Mit steigender Komplexität ontologiegestützter Modelle sinkt tendenziell ihre Nachvollziehbarkeit für Modellbenutzer ohne eine informationstechnische Unterstützung. Damit ein Modellierungskonzept informationstechnische Unterstützung erfahren kann, muss es *implementierbar* sein. Der Begriff der Implementierbarkeit eines Modellierungskonzepts umfasst die informationstechnische Verarbeitbarkeit *aller* seiner notwendigen Bestandteile. In der Regel setzt eine solche informationstechnische Verarbeitbarkeit eine *formale* Spezifikation des Modellierungskonzepts voraus. Formale Spezifikationen können nämlich grundsätzlich in eine computerinterne Repräsentation überführt werden. Voraussetzung hierfür ist die Abbildbarkeit der verwendeten Ausdrucksmittel auf die dazu korrespondierenden Konstrukte der ausgewählten Programmiersprache.[2)] Die Implementierbarkeit des Modellierungskonzepts steigt somit mit dem Anteil programmiersprachlicher Konstrukte, auf die das Modellierungskonzept abgebildet werden kann. Insofern ist die Implementierbarkeit eines Modellierungskonzepts relativ zu einer Programmiersprache zu beurteilen.

Anhand der computerinternen Repräsentation sowohl statischer als auch dynamischer Phänomene können computergestützte Auswertungs- und Simulationsverfahren durchgeführt werden. Die Auswertung umfasst alle Analysen von Modellen bezüglich bestimmter Eigenschaften. Beispielsweise kann es von Interesse sein zu analysieren, ob bestimmte Zustände eines ontologiegestützten Modells durch modellendogene Operationsausführungen erreicht werden können. Als solche Zustände kommen z.B. intendierte Zielzustände, in denen eine vorgegebene Aufgabe erfüllt ist, oder so genannte „Home States“, aus denen das modellierte System stets ein Operation in wohldefinierter Weise von Neuem ausführen kann, in Frage. Bei der Simulation hingegen wird u.a. das Eintreten von Ereignissen computergestützt nachgeahmt. Die hierbei gewonnenen Erkenntnisse können möglicherweise auf den repräsentierten Realitätsausschnitt übertragen werden.[3)]

---

1) Vgl. SOWA (1984), S. 69 ff.; SOWA (2000), S. 476 ff. Vgl. darüber hinaus HANSEN ET AL. (1992), S. 119 ff. zur Überführung von Conceptual Graphs („begriffliche Graphen“; HANSEN ET AL. (1992), S. 24) in Entity-Relationship-Diagramme.

2) Vgl. FRANK/VAN LAAK (2003), S. 33.

3) Vgl. GRUHN (1996), S. 99; OBERWEIS (1996), S. 210 f.; SCHMID (1998), S. 4 ff.; ZIMMER (2001), S. 14.

2.2 Formaler Rahmen des integrativen Modellierungskonzepts

2.2.1 Prädikatenlogik

2.2.1.1 Konventionelle Prädikatenlogik

2.2.1.1.1 Überblick über die konventionelle Prädikatenlogik

Der formale Rahmen des integrativen Modellierungskonzepts wird in erster[1)] Linie von der *Prädikatenlogik erster Ordnung*[2)] gestellt. Sie wird als formaler Ausgangspunkt für *ontologische Signaturen* verwendet, welche als Komponenten von Ontologien vorgestellt werden. Zu diesem Zweck wird mit steigender Komplexität von der konventionellen über die sortierte Prädikatenlogik schließlich zu Ontologien übergegangen.

Die Verwendung der Prädikatenlogik als formaler Ausgangspunkt für Ontologien ist aus mehrfacher Perspektive viel versprechend: Erstens liegt bislang keine formale Konzeption für Ontologien vor, die keinen – zumindest *mittelbaren* – Bezug zur Prädikatenlogik erkennen lässt. Sämtliche „belastbaren" theoretischen Ausarbeitungen zu Ontologien bedienen sich entweder unmittelbar der Prädikatenlogik oder lassen sich „verlust-

1) Komplementär zu der Prädikatenlogik werden im weiteren Verlauf *Multimengen* vorgestellt. Allerdings fällt die Bedeutung von Multimengen für das gesamte Konzept deutlich geringer aus als die der Prädikatenlogik.

2) Für die vorliegende Arbeit wird lediglich das Kalkül der Prädikatenlogik *erster* Ordnung benötigt. Daher wird im Folgenden der Zusatz „erster Ordnung" weggelassen. Die Beschränkung auf die Prädikatenlogik erster Ordnung äußert sich darin, dass die – noch vorzustellenden – Quantoren und Relationssymbole sich lediglich auf – ebenso noch vorzustellende – Terme beziehen dürfen. Dieser Einschränkung des Ausdrucksvermögens liegt eine stufenweise Unterscheidung von formalen Objekten zugrunde. Während die Elemente (Terme) des Grundbereichs als formale Objekte *erster* Ordnung behandelt werden, gelten Relations- und Operationssymbole als Objekte *zweiter* Stufe. Durch den Übergang auf die Prädikatenlogik *zweiter* Ordnung wird der Bezug von Quantoren und Relationssymbolen zu Objekten zweiter Stufe erlaubt; vgl. MANZANO (1993), S. 46 ff. Dadurch lassen sich beispielsweise Eigenschaften von Relationssymbolen explizit ausdrücken, die in einer Prädikatenlogik erster Ordnung lediglich implizit enthalten sein können. Dieser Aspekt der Prädikatenlogik zweiter Ordnung erweist sich im Kontext von Ontologien als viel versprechend; vgl. GUARINO/WELTY (2000), S. 101 ff. für den Hinweis auf „Meta-Prädikate", die zur Charakterisierung von Konzepten in einer Ontologie verwendet werden können; sie entsprechen den zuvor erwähnten *Meta-Konzepten*. Allerdings wird der mit einem Übergang zur Prädikatenlogik zweiter Ordnung verbundene Komplexionssprung vom Verfasser als zu hoch eingeschätzt, als dass die Vorteile des Zugewinns an Ausdrucksvermögen gerechtfertigt wären. Darüber hinaus ist mit dem Übergang zur Prädikatenlogik zweiter Ordnung der Verlust beweistheoretischer Eigenschaften – insbesondere der Vollständigkeit – der Prädikatenlogik erster Ordnung verbunden; vgl. HERMES (1991), S. 141 u. 155 ff.; KOHLHASE (1992), S. 226 ff.

frei“ mit der Prädikatenlogik in Einklang bringen.[1] Auch die vorliegende Arbeit bedient sich der prädikatenlogischen Systemrepräsentation, indem Ontologien als Erweiterungen prädikatenlogischer Spezifikationen charakterisiert werden.

Zweitens liegen mit höheren Petri-Netzen bereits Techniken vor, die einen – zumeist *unmittelbaren* – Bezug zur Prädikatenlogik aufweisen. Petri-Netze wurden bereits in der Einleitung der vorliegenden Arbeit als ein Baustein für das integrative Modellierungskonzept vorausgesetzt. In der Variante der Prädikat/Transition-Netze werden z.B. prädikatenlogische Ausdrücke dazu verwendet, die topologischen Komponenten von Petri-Netzen zu beschriften.[2]

Drittens weist die Prädikatenlogik eine bemerkenswerte Nähe zu implementierungsnahen Formalismen auf. Zum Großteil ist eine Abwärtskompatibilität prädikatenlogischer Ausdrücke mit programmiersprachlichen Ausdrücken gegeben.[3] Dies trifft in erster Linie für die Programmiersprache PROLOG zu, die sich seit längerem im Umfeld der Künstlichen-Intelligenz-Forschung durchgesetzt hat.[4] Dabei hat sich die Programmiersprache PROLOG in den letzten Jahren als so robust erwiesen, dass sie trotz unterschiedlichster Erweiterungen das Spektrum prädikatenlogischer Wissensrepräsentationen zum Großteil zu erfassen vermag. Beispielsweise liegen bereits mehrere PROLOG-Dialekte vor, die eine unmittelbare Implementierung von prädikatenlogischen Wissensrepräsentationen inklusive *sortierter Ausdrücke*[5] erlauben.[6]

In einer weiteren Funktion wird die Prädikatenlogik dazu verwendet, mit jedem Übergang zu einer weiteren Unterart der Prädikatenlogik die Semantik der hinzukommenden Konstrukte zu präzisieren. Bei den hinzukommenden Konstrukten handelt es sich zumeist um metasprachliche Ausdrucksmittel, mit denen metasprachliche Ausdrücke konstruiert werden. Bei der Konstruktion dieser metasprachlichen Ausdrücke wird wiederum auf die Prädikatenlogik zurückgegriffen. Somit wird mit der vorliegenden Arbeit ei-

---

1) Vgl. GRUBER (1993), S. 203; GUARINO (1995), S. 631 ff.; GUARINO (1997), S. 298 f.; SMITH (1996), S. 289 ff.; SOWA (2000A) S. 70 ff.; USCHOLD/GRUNINGER (1996), S. 109 f.; WELTY/GUARINO (2001), S. 56 ff.

2) Vgl. GENRICH (1987), S. 212 ff.; KLEINJOHANN (1993), S. 59 ff.; REISIG (1991), S. 122 ff.; TACKEN (2000), S. 42 ff.

3) Vgl. BIBEL (1992), S. 186 ff.; VAN EMDEN/KOWALSKI (1976), S. 733 ff.

4) Vgl. DAS (1992), S. 211 ff.; GALLIER (1986), S. 410 ff.

5) Sortierte Ausdrücke werden in Abschnitt 2.2.1.2.2.2 in Form sortierter Terme und sortierter Formeln vorgestellt.

6) Ausnahmen hiervon bilden beispielsweise so genannte Horn-Klauseln, auf die PROLOG bisweilen beschränkt ist. Horn-Klauseln werden in Abschnitt 3.1.4.1 näher vorgestellt.

nem *induktiven*[1)] Definitionsschema gefolgt. Während bei der Einführung der konventionellen Prädikatenlogik noch größtenteils eine natürlichsprachliche Erklärung der sprachlichen Konstrukte erfolgt, wird bei den Erklärungen der sprachlichen Konstrukte aus der sortierten Prädikatenlogik vermehrt auf die konventionelle Prädikatenlogik zurückgegriffen. Dadurch weist der gesamte Ansatz eine Reduzierbarkeit auf die Prädikatenlogik auf, wobei die Ausdrücke unterschiedlichen Sprachebenen zugehören können. Diese Reduzierbarkeit auf die Prädikatenlogik hat eine Abgeschlossenheit des integrativen Modellierungskonzepts gegenüber „artfremden" Formalismen zufolge.

2.2.1.1.2 Syntaktische Aspekte der konventionellen Prädikatenlogik

2.2.1.1.2.1 Konventionelle Signaturen

Bei den syntaktischen Aspekten der konventionellen Prädikatenlogik handelt es sich einerseits um ein *Alphabet* und andererseits um eine *Grammatik* zur Konstruktion von Ausdrücken.[2)] Die Grammatik umfasst Formierungsregeln, nach denen Ausdrücke mittels Elementen des prädikatenlogischen Alphabets konstruiert werden können. In diesem Punkt unterscheiden sich konventionelle und sortierte prädikatenlogische Sprachen. Die Formierungsregeln für Ausdrücke fallen nämlich in den beiden Varianten der Prädikatenlogik unterschiedlich aus.

Das konventionelle Alphabet $ALPH_{KS}$ einer prädikatenlogischen Sprache erster Ordnung umfasst:[3)]

(1.) die logischen Symbole[4)]
Konnektoren

---

1) Induktive Definitionsschemata zeichnen sich dadurch aus, dass aus jeweils bekannten (Basis-)Konstrukten und Regeln zur Konstruktion von Konstrukten neue (zusammengesetzte) Konstrukte erstellt werden; vgl. ZELEWSKI (1995), Bd. 4, S. 40. Die bereits bekannten Konstrukte werden als Induktionsbasis der induktiven Definition vorausgesetzt. In Bezug auf die vorliegende Arbeit ist die natürliche Sprache die Induktionsbasis. Es wird ein intuitives Vorverständnis für natürlichsprachliche Konstrukte vorausgesetzt, um formalsprachliche Konstrukte induktiv definieren zu können. An anderen Stellen werden formale Konstrukte induktiv durch andere formale Konstrukte definiert. Letztgenannte formale Konstrukte werden dann als Induktionsbasis bezeichnet, obwohl eine weitere Vertiefung der Definition unweigerlich zur natürlichen Sprache führen muss.
Induktive Definitionsschemata werden in den Wirtschaftswissenschaften oftmals in Anspruch genommen. Sie werden teilweise synonym als *Definitionsgleichungen* bezeichnet. In Definitionsgleichungen werden *endogene* Variablen durch *exogene* Variablen erklärt. Die exogenen Variablen sind dann die Basiskonstrukte, mit denen die zusammengesetzten Konstrukte – nämlich die endogenen Variablen – erklärt werden.

2) Vgl. ZELEWSKI (1995), Bd. 4, S. 95.

3) Vgl. GALLIER (1986), S. 147; EBBINGHAUS ET AL. (1992), S. 17; ZELEWSKI (1995), Bd. 4, S. 93 f.

4) Die Bezeichnungen *Symbol* und *Zeichen* werden fortan synonym verwendet.

Konjunktor	$\wedge$
Adjunktor	$\vee$
Disjunktor	$\underline{\vee}$
Subjunktor	$\rightarrow$
Bijunktor	$\leftrightarrow$
logische Operatoren	
Negator	$\neg$
Quantoren	
Allquantor	$\forall$
Existenzquantor	$\exists$
Einsquantor	$\underline{\exists}$

(2.) deskriptive Symbole

Operationssymbole	O_i mit $i = 1,...,I$, $I \in \mathbb{N}_+$ und $OPS = \{O_i\}$
Relationssymbole	R_j mit $j = 1,...,J$, $J \in \mathbb{N}_+$ und $RS = \{R_j\}$
Variablen	x_q mit $q = 1,...,Q$, $Q \in \mathbb{N}_+$ und $VAR = \{x_q\}$

(3.) Hilfssymbole

$( ) : , < > .$

Die Elemente der Menge $ALPH_{KS}$ werden benötigt, um eine prädikatenlogische *Sprache* zu spezifizieren. Letztgenanntes entspricht der Menge aller Symbole aus dem konventionellen Alphabet $ALPH_{KS}$ und der Menge aller Formierungsregeln für zulässige prädikatenlogische Ausdrücke.

Um die Zulässigkeit prädikatenlogischer Ausdrücke bestimmen zu können, werden die Operations- und Relationssymbole aus einem konventionellen Alphabet $ALPH_{KS}$ in einer konventionellen Signatur SIG_{KS}[1] mit Hilfe von Typisierungsfunktionen *typisiert*. Durch die Typisierung von Operations- und Relationssymbolen werden metasprachliche Integritätsbedingungen formuliert, denen zulässige Ausdrücke genügen müssen, die mit den Symbolen aus dem konventionellen Alphabet $ALPH_{KS}$ konstruiert sind. Darüber hinaus dient das Alphabet $ALPH_{KS}$ als Grundlage sowohl für *sortierte* als auch für *ontologische Signaturen*. Zu diesem Zweck wird im weiteren Verlauf das konventionelle Alphabet $ALPH_{KS}$ um eine Menge von *Sorten* bzw. *Konzepten* erweitert.

Die Menge der logischen Symbole aus dem konventionellen Alphabet $ALPH_{KS}$ weist in der oben aufgeführten Form *Redundanzen* auf.[2] Die Redundanz ist bezüglich *zusam-*

---

1) Auf konventionelle Signaturen wird im Folgenden ausführlich eingegangen.

2) Die aufgeführten logischen Symbole werden zudem in verschiedenen Quellen teilweise unterschiedlich notiert und teilweise ausgelassen. Die unterschiedlichen Notationen sind größtenteils historisch bedingt. Das Auslassen von Symbolen ist auf ihre – bereits oben erwähnte – Redundanz zurückzuführen.

mengesetzter Formeln gegeben, für deren Aufbau logische Symbole benötigt werden. Denn es ließen sich mit nur einem Konnektor, dem Negator und nur einem Quantor alle zusammengesetzten Formeln *äquivalent*[1)] ausdrücken, die mit den oben angegebenen Symbolen und den im Folgenden vorzustellenden Formierungsregeln für Formeln konstruierbar sind.[2)] Die redundanten logischen Symbole erlauben allerdings eine *kompaktere* Formelkonstruktion. Daher werden sie im Folgenden beibehalten. Auf die äquivalenten Ausdrucksweisen wird später – im Kontext der prädikatenlogischen Semantik – eingegangen.[3)]

Konventionelle[4)] *Ausdrücke* werden mit Hilfe der Symbole aus dem konventionellen Alphabet $ALPH_{KS}$ konstruiert. Sämtliche konventionellen Ausdrücke sind Elemente des Monoids $ALPH_{KS}$*[5)]. Die Menge $ALPH_{KS}$* umfasst alle *Wörter*, die mit den Symbolen aus dem Alphabet $ALPH_{KS}$ konstruiert werden können. Dabei ist jede endliche, eventuell leere Folge von Symbolen aus dem Alphabet $ALPH_{KS}$ ein *Wort* über $ALPH_{KS}$.[6)] So-

---

1) Die *Äquivalenz* von Formeln wird im weiteren Verlauf formal präzisiert. Bis zu dieser formalen Präzisierung wird ein intuitives Verständnis für die Äquivalenz von Ausdrücken i.S.v. „bedeutungsgleich" vorausgesetzt.

2) Vgl. ROTHMALER (1995), S. 32 ff.

3) Vgl. Abschnitt 2.2.1.1.3.

4) Neben *konventionellen* Ausdrücken werden in der vorliegenden Arbeit auch *sortierte* und *ontologische* Ausdrücke thematisiert. Wenn aus dem Argumentationskontext hervorgeht, um welche Ausdrucksart es sich handelt, wird der entsprechende Zusatz ausgelassen.

5) Der *Kleene-Stern* „*" findet in der vorliegenden Arbeit mehrfach Verwendung. Er wird dazu verwendet, ausgehend von einer Referenzmenge – in dem o.a. Fall ALPH – einen *Monoid* zu konstruieren. Bei dem Monoid ALPH* zu der Menge ALPH handelt es sich um ein Drei-Tupel $(S,+,\lambda)$. Die Menge S wird als „Referenzmenge" bezeichnet. + ist eine binäre Operation, die Zwei-Tupel bestehend aus Elementen der Menge S auf die Menge S* abbildet. λ ist ein ausgezeichnetes Element der Menge S*.

   Monoide zeichnen sich durch drei Eigenschaften aus. Erstens sind Monoide *abgeschlossen* bezüglich des Verkettungsoperators +:

$$\forall s_1,s_2 \in S^*: (s_1+s_2) \in S^*.$$

   Zweitens ist das ausgezeichnete Element λ neutral bezüglich der Operation +:

$$\forall\ s \in S^*: \lambda+s = s+\lambda = s.$$

   Drittens ist die Operation + assoziativ:

$$\forall s_1,s_2,s_3 \in S^*: (s_1+s_2)+s_3 = s_1+(s_2+s_3).$$

   Wird kein neutrales Element berücksichtigt, handelt es sich um eine *Halbgruppe*.
   Bei dem Monoid S* wird die Verknüpfungsoperation + in der Regel nicht explizit angegeben. Dieser Vorgehensweise wird auch in der vorliegenden Arbeit gefolgt.

6) Zur Definition von *Wörtern* über einem formalsprachlichen Alphabet vgl. EHRIG ET AL. (1999), S. 25; ERK/ PRIESE (2002), S. 27.

mit handelt es sich auch bei allen konventionellen Ausdrücken, die über dem konventionellen Alphabet $ALPH_{KS}$ konstruierbar sind, um Wörter.

Die Menge der deskriptiven Symbole umfasst die Menge OPS aller Operationssymbole, die Menge RS aller Relationssymbole und die Menge VAR aller Variablen. Die Bezeichnungen *Operations*- und *Relationssymbole* sind dadurch motiviert, dass die entsprechenden Elemente in einer formalen Semantik durch *Operationen* bzw. *Relationen* interpretiert werden. Synonym zu Operations- und Relationssymbol werden auch die Bezeichnungen *Funktions*- bzw. *Prädikatssymbol* zugelassen.

Während die Menge der logischen Symbole aus dem konventionellen Alphabet $ALPH_{KS}$ *fix* ist, sind die deskriptiven Symbole *variabel*. Die Elemente der Mengen OPS, RS und VAR können nämlich beliebig spezifiziert werden. Aufgrund der Variabilität der deskriptiven Symbole fällt das konventionelle Alphabet $ALPH_{KS}$ potenziell unendlich aus.[1)]

Zudem ist in einem konventionellen Alphabet $ALPH_{KS}$ zunächst ungeklärt, welchen symbolspezifischen *Integritätsbedingungen* Ausdrücke folgen müssen, die mit Komponenten des Alphabets $ALPH_{KS}$ konstruiert werden können. Solche symbolspezifischen Bedingungen werden erst durch die *Typisierung* der Operations- und Relationssymbole in einer konventionellen Signatur SIG_{KS} ausgedrückt.

Eine konventionelle Signatur SIG_{KS} ist definiert als das Fünf-Tupel:

$$SIG_{KS} = (OPS,RS,VAR,typ_{OPS_{KS}},typ_{RS_{KS}}).$$

Die Komponenten einer konventionellen Signatur sind:

(1.) Eine Menge $OPS = (O_1,...,O_I)$ von *Operationssymbolen* O_i mit $i=1,...,I$ und $I \in \mathbb{N}$,

(2.) eine Menge $RS = (R_1,...,R_J)$ von *Relationssymbolen* R_j mit $j=1,...,J$ und $J \in \mathbb{N}$,

(3.) eine Menge $VAR = (x_1,...,x_Q)$ von *Variablen* x_q mit $q = 1,...,Q$ und $Q \in \mathbb{N}$,

(4.) eine Operationssymbol-Typisierungsfunktion $typ_{OPS_{KS}}$: $OPS \rightarrow \mathbb{N}$ und

(5.) eine Relationssymbol-Typisierungsfunktion $typ_{RS_{KS}}$: $RS \rightarrow \mathbb{N}_+$.

Bei den Elementen der Operationssymbolmenge OPS, der Relationssymbolmenge RS und der Variablenmenge VAR handelt es sich um *objektsprachliche* Konstrukte. Mit ihnen können objektsprachliche *Ausdrücke* über einer konventionellen Signatur SIG_{KS} konstruiert werden. Dabei wird von den Mengen OPS, RS und VAR angenommen, dass sie paarweise disjunkt sind. Somit gilt:

$$OPS \cap RS = OPS \cap VAR = RS \cap VAR = \varnothing.$$

Die Typisierungsfunktionen $typ_{OPS_{KS}}$ und $typ_{RS_{KS}}$ sind hingegen *metasprachliche* Konstrukte. Sie werden dazu verwendet, die objektsprachlichen Operations- bzw. Relations-

1) Vgl. ZELEWSKI (1995), Bd. 4, S. 96.

symbole zu typisieren. Jede einzelne Typisierung hat die Qualität eines metasprachlichen Ausdrucks, da eine Aussage über objektsprachliche Konstrukte getätigt wird.

Die metasprachliche Typisierung schränkt die Menge der Ausdrücke, die in der Objektsprache konstruiert werden können, ein. Jedem Operations- und Relationssymbol wird nämlich durch die Typisierungsfunktionen $typ_{OPS_{KS}}$ bzw. $typ_{RS_{KS}}$ eine *Stelligkeit* zugewiesen, die bei der Konstruktion von Ausdrücken berücksichtigt werden muss. Die Stelligkeit eines Operations- bzw. Relationssymbols gibt die Anzahl von *Termen*[1)] an, über die sich das Symbol in seinem Argument erstrecken kann, damit der dadurch entstehende Ausdruck zulässig ist. Für ein Operationssymbol $O_i \in OPS$ wird beispielsweise durch $typ_{OPS_{KS}}(O_i)=m$ mit $m \in \mathbb{N}$ angegeben, dass es genau m Terme in seinem Argument aufnehmen kann. Dabei werden auch Operationssymbole $O_i \in OPS$ mit der Stelligkeit $typ_{OPS_{KS}}(O_i)=0$ zugelassen. Solche null-stelligen Operationssymbole werden als *Konstantensymbole* bezeichnet.

Die Menge $KON_{SIG_{KS}}$ aller Konstantensymbole bildet gemeinsam mit der Menge VAR die Menge

$$IND_{SIG_{KS}} = KON_{SIG_{KS}} \cup VAR$$
$$\text{mit } KON_{SIG_{KS}} = \{ O_i \mid O_i \in OPS \wedge typ_{OPS_{KS}}(O_i)=0\}$$
$$KON_{SIG_{KS}} \cap VAR = \varnothing$$

aller *Individuensymbole* über einer konventionellen Signatur SIG_{KS}. Die Menge $IND_{SIG_{KS}}$ ist spezifisch für eine konventionelle Signatur SIG_{KS} definiert. Die Signaturspezifität der Menge $IND_{SIG_{KS}}$ ergibt sich aus der Signaturspezifität ihrer Teilmenge $KON_{SIG_{KS}}$. Konstantensymbole gehen nämlich erst aus der signaturspezifischen Typisierung von Operationssymbolen mit der Zahl 0 hervor.

Die Elemente der Menge VAR sind Variablen. In ihrer Funktionalität sind Variablen Konstantensymbolen grundsätzlich sehr ähnlich. Die Elemente beider Mengen werden nämlich später extensional durch formale Objekte interpretiert. Während Konstantensymbole *fixe* Größen sind, die nicht variiert werden können, haben Variablen per definitionem einen *parametrischen* Charakter in Bezug auf die Interpretation von Ausdrücken über einer konventionellen Signatur SIG_{KS}. Für alle Ausdrücke über einer konventionellen Signatur SIG_{KS} können nämlich eindeutige Interpretationen bestimmt werden, solange sie nicht auf Variablen zurückgreifen. Sobald Variablen verwendet werden, ist zudem eine *Wertezuweisung* für die Variablen notwendig. Die Wertezuweisung für die Variablen wird auch als deren *Belegung* bezeichnet. Die Interpretation von Ausdrücken, die Variablen enthalten, erfolgt dann unter der Berücksichtigung der parametrischen Belegung der darin frei vorkommenden Variablen.

Üblicherweise werden Variablen in konventionellen Signaturen nicht aufgeführt.[2)] Dieser Vorgehensweise könnte die Begründung zugrunde liegen, dass Variablen sprachli-

1) Terme werden im nächsten Abschnitt vorgestellt.

2) Vgl. z.B. BEIERLE/KERN-ISBERNER (2000), S. 47.

che Komponenten sind, die nicht näher bestimmt zu werden brauchen, da es sich bei ihnen um domänenunabhängige sprachliche Konstrukte handelt. Der Verfasser präferiert allerdings die explizite Aufzählung aller benötigten Variablen, um später die Komponenten von Ausdrücken über konventionellen Signaturen formal eindeutig charakterisieren zu können. Zudem handelt es sich auch bei den logischen Symbolen aus einer Signatur um domänenunabhängige Zeichen. Darüber hinaus wird eine Anschlussfähigkeit konventioneller Signaturen sowohl an sortierte als auch an ontologische Signaturen bewahrt. In beiden Fällen ist es nötig, Variablen den jeweils verwendeten Strukturierungseinheiten – Sorten oder Konzepten – zuzuordnen. Entsprechend müssen sowohl in konventionellen als auch in ontologischen Signaturen Variablen in der Signatur aufgeführt werden.

Für ein Relationssymbol $R_j \in RS$ wird durch $typ_{RS_{KS}}(R_j)=n$ angegeben, dass es genau n Terme in seinem Argument aufnehmen kann. Relationssymbole können nur mit einer positiven Zahl $n \geq 1$ typisiert werden.[1)]

In der Regel wird auf die signaturbezogene Charakterisierung deskriptiver Symbole aus einem Alphabet $ALPH_{KS}$ bei konventionellen prädikatenlogischen Sprachen verzichtet.[2)] Entweder wird dann auf eine explizite Angabe der Stelligkeiten von Operations- und Relationssymbolen ganz verzichtet, oder die Stelligkeiten werden als Indizes an die jeweiligen Symbole angehängt. Die signaturbezogene Charakterisierung deskriptiver Symbole wird hier allerdings aus mehreren Gründen bevorzugt. Erstens ist jede signaturbezogene Charakterisierung deskriptiver Symbole „abwärtskompatibel“ zu jeder alternativen Darstellungsform. Die Bilder der Typisierungsfunktionen $typ_{OPS_{KS}}$ und $typ_{RS_{KS}}$ lassen sich nämlich mühelos zu Indizes der jeweiligen Symbole transformieren. Für ein Operationssymbol $O_i \in OPS$ mit der Typisierung $typ_{OPS_{KS}}(O_i)=n$ wird dann O_i^n notiert. Analog wird R_j^m für ein Relationssymbol $R_j \in RS$ mit $typ_{RS_{KS}}(R_j)=m$ notiert. Es gilt daher die Äquivalenz:

$$O_i \in OPS \wedge typ_{OPS_{KS}}(O_i)=m :\Leftrightarrow O_i^m,$$

$$R_j \in RS \wedge typ_{RS_{KS}}(R_j)=n :\Leftrightarrow R_j^n.$$

Somit umfasst die signaturbezogene Charakterisierung die alternative Darstellungsform mit Indizes.

Zweitens ist mit der vorliegenden Arbeit das Bestreben verbunden, eine induktive Argumentationslinie durchzuhalten. Die Induktionsbasis der Argumentation ist die konventionelle Prädikatenlogik. Die Spezifikation von deskriptiven Symbolen in einer konventionellen Signatur SIG_{KS} trägt dazu bei, die Argumentationsweise formal und somit auch didaktisch präziser durchzuhalten. Für die deskriptiven Symbole wird nämlich im Rahmen der sortierten Prädikatenlogik im Gegensatz zu den deskriptiven Symbolen aus der konventionellen Prädikatenlogik stets eine Typisierung innerhalb einer Signatur

1) Dadurch wird der aussagenlogische Grenzfall strukturloser Relationssymbole ausgeschlossen.

2) Vgl. z.B. EBBINGHAUS ET AL. (1992), S. 16 f. GALLIER (1986), S. 147 ff.

vorausgesetzt. Darüber hinaus werden auch die deskriptiven Symbole, die für die Konstruktion einer Ontologie benötigt werden, innerhalb einer *ontologischen Signatur* typisiert.

In Anlehnung an die Schreibweise[1)] für Operationen und Relationen werden Operations- und Relationssymbole entsprechend dem folgenden Schema für die Spezifikation konventioneller Signaturen deklariert:

SIG_{KS}

OPS:

$O_1{:}\rightarrow m_1$

...

$O_I{:}\rightarrow m_I$

RS:

$R_1{:}\rightarrow n_1$

...

$R_J{:}\rightarrow n_J$

VAR:

$x_1,\ldots,x_Q$

2.2.1.1.2.2 Konventionelle Ausdrücke

2.2.1.1.2.2.1 Konventionelle Terme

Über einer konventionellen Signatur SIG_{KS} können zwei Arten von Ausdrücken konstruiert werden. Es handelt sich hierbei um *konventionelle Terme* und *konventionelle Formeln*. Zur Konstruktion von konventionellen Formeln werden konventionelle Terme benötigt. Daher werden im Folgenden zunächst konventionelle Terme vorgestellt, um im Anschluss daran auf konventionelle Formeln eingehen zu können.

Die Menge $EXPR_{SIG_{KS}}$ der *konventionellen Ausdrücke* über einer konventionellen Signatur SIG_{KS} ist eine Teilmenge der Menge $ALPH_{KS}$* aller Wörter über dem konventionellen Alphabet $ALPH_{KS}$:

$$EXPR_{SIG_{KS}} \subset ALPH_{KS}{}^{*}.$$ [2)]

1) Vgl. EHRIG ET AL. (1999), S. 312.

2) Die Gleichheit ($EXPR_{SIG_{KS}} = ALPH_{KS}{}^{*}$) ist dadurch ausgeschlossen, dass in der Menge $ALPH_{KS}$* alle denkbaren Zeichenketten enthalten sind, die über dem konventionellen Alphabet $ALPH_{KS}$ enthalten sind. Hierin sind auch alle Zeichenketten enthalten, die als konventionelle Ausdrücke nicht zugelassen sind, da sie nicht den Formierungsregeln für konventionelle Ausdrücke genügen.

Die Menge $EXPR_{SIG_{KS}}$ setzt sich zum einen aus der Menge $TERM_{SIG_{KS}}$ aller konventionellen Terme und zum anderen aus der Menge $FORM_{SIG_{KS}}$ aller konventionellen Formeln über SIG_{KS} zusammen:

$$EXPR_{SIG_{KS}} = TERM_{SIG_{KS}} \cup FORM_{SIG_{KS}}$$
$$\text{mit } TERM_{SIG_{KS}} \cap FORM_{SIG_{KS}} = \varnothing$$

Die induktive Grammatik zur Konstruktion von konventionellen Termen lautet:[1)]

(1.) Wenn $x_q \in VAR$ mit $q \in \{1,...,Q\}$ und $Q \in \mathbb{N}$ gelten, dann gilt auch $x_q \in TERM_{SIG_{KS}}$. Jede Variable $x_q \in VAR$ ist ein *atomarer konventioneller Term*.

(2.) Wenn $O_i \in OPS$ und $typ_{OPS_{KS}}(O_i)=0$ und somit auch $O_i \in KON_{SIG_{KS}}$ gelten, dann gilt auch $O_i \in TERM_{SIG_{KS}}$. Jedes Konstantensymbol ist ein *atomarer konventioneller Term*.[2)]

(3.) Wenn $O_i \in OPS$ und $typ_{OPS_{KS}}(O_i)=m$ mit $m \in \mathbb{N}_+$ und $t_1,...,t_m \in TERM_{SIG_{KS}}$ gelten, dann gilt auch $O_i(t_1,...,t_m) \in TERM_{SIG_{KS}}$. $O_i(t_1,...,t_m)$ ist dann ein *zusammengesetzter konventioneller Term*.

Für die Definition von konventionellen Termen sind lediglich Variablen und Operationssymbole von Bedeutung. Relationssymbole finden in der Definition von konventionellen Termen keine Berücksichtigung. Atomare konventionelle Terme sind nämlich entweder Variablen oder sind null-stellige Operationssymbole (Konstantensymbole). Zusammengesetzte konventionelle Terme gehen hingegen aus der typgerechten Anwendung eines Operationssymbols auf andere konventionelle Terme hervor.

Bei allen Variablen aus einer konventionellen Signatur SIG_{KS} handelt es sich um konventionelle Terme. Somit ist die Menge VAR aller Variablen aus dem konventionellen Alphabet $ALPH_{KS}$ eine Teilmenge[3)] der Menge $TERM_{SIG_{KS}}$ aller konventionellen Terme über einer Signatur SIG_{KS}:

$$VAR \subseteq TERM_{SIG_{KS}}.$$

---

1) Vgl. EBBINGHAUS ET AL. (1992), S. 18; ROTHMALER (1995), S. 30; SCHÖNING (1992), S. 52; ZELEWSKI (1995), Bd. 4, S. 94.

2) Für atomare konventionelle Terme, die aus Konstantensymbolen hervorgehen, wird verkürzt die Schreibweise O_i anstelle von $O_i()$ zugelassen. Die zweite Schreibweise ist zwar kompatibel mit der Schreibweise für zusammengesetzte konventionelle Terme führt allerdings zu einer unnötig verkomplizierten Schreibweise.

3) Es wird hier der allgemeingültige Fall dargestellt, in dem die Menge VAR der Variablen aus einer konventionellen Signatur SIG_{KS} eine *unechte* Teilmenge der Menge $TERM_{SIG_{KS}}$ aller konventionellen Terme über SIG_{KS} ist. Dadurch wird auch der Fall umfasst, bei dem die Menge OPS aller Operationssymbole leer ist. In diesem Fall kämen nur Variablen als konventionelle Terme in Frage. Entsprechend müsste für eine konventionelle Signatur SIG_{KS} mit zugrunde liegender Menge $OPS=\varnothing$ von Operationssymbolen gelten:

$$VAR = TERM_{SIG_{KS}}.$$

Bei konventionellen Termen, die aus Operationssymbolen hervorgehen, können zwei Arten unterschieden werden. Bei der ersten Art von konventionellen Termen handelt es sich um Konstantensymbole. Terme dieser Art werden von der o.a. Regel (2.) erfasst. Die Menge $KON_{SIG_{KS}}$ aller Konstantensymbole ist somit eine Teilmenge der Menge $TERM_{SIG_{KS}}$ aller konventionellen Terme:

$$KON_{SIG_{KS}} \subseteq TERM_{SIG_{KS}}$$

Hierbei gilt die Prämisse, dass atomare konventionelle Terme, die aus Konstantensymbolen hervorgehen, in der Form O_i angegeben werden können. Die Anwendung $O_i()$ eines null-stelligen Operationssymbols O_i auf das leere Argument wird durch $TERM_{SIG_{KS}}$ nicht umfasst.

Entsprechend ist auch die Menge $IND_{SIG_{KS}}$ aller Individuensymbole über einer konventionellen Signatur SIG_{KS} eine Teilmenge der Menge $TERM_{SIG_{KS}}$:

$$IND_{SIG_{KS}} \subseteq TERM_{SIG_{KS}}.$$

Die zweite Art von Termen, die aus Operationssymbolen hervorgehen kann, wird von der o.a. Regel (3.) erfasst. Es handelt sich hierbei um konventionelle Terme, die aus der Anwendung eines Operationssymbols O_i auf ein m-Tupel $(t_1,...,t_m)$ von Termen hervorgehen. Damit ein Ausdruck $O_i(t_1,...,t_m)$ ein konventioneller Term sein kann, muss $typ_{OPS_{KS}}(O_i)=m$ gelten. Das heißt, um mit Hilfe eines Operationssymbols O_i einen konventionellen Term konstruieren zu können, muss die natürliche Zahl m berücksichtigt werden, die als Typ für O_i angegeben ist. Nur in diesem Fall ist eine typgerechte Anwendung des Operationssymbols auf ein Termtupel gewährleistet. Ein Operationssymbol O_i kann bei der typgerechten Konstruktion eines konventionellen Terms nur auf so viele konventionelle Terme angewendet werden, wie in der entsprechenden Typisierung $typ_{OPS_{KS}}(O_i)$ angegeben ist.

Konventionelle Terme über einer Signatur SIG_{KS} können hinsichtlich zweier Kriterien *klassifiziert* werden. Das erste Kriterium zur Klassifikation von Termen betrifft ihre – bereits oben erwähnte – *Zusammengesetztheit*. Die Menge $TERM_{SIG_{KS}}$ aller Terme über einer konventionellen Signatur SIG_{KS} wird entsprechend dem Kriterium der Zusammengesetztheit in die zwei zueinander disjunkten Teilmengen $AT_{SIG_{KS}}$ und $ZT_{SIG_{KS}}$ unterteilt:

$$TERM_{SIG_{KS}} = AT_{SIG_{KS}} \cup ZT_{SIG_{KS}}$$
$$\text{mit } AT_{SIG_{KS}} \cap ZT_{SIG_{KS}} = \varnothing$$

Die Menge

$$AT_{SIG_{KS}} = \{t \mid t \in \{VAR \cup KON_{SIG_{KS}}\}\}$$
$$\text{oder } AT_{SIG_{KS}} = \{t \mid t \in IND_{SIG_{KS}}\}$$

der *atomaren konventionellen Terme* über einer konventionellen Signatur SIG_{KS} stimmt mit der Menge $IND_{SIG_{KS}}$ aller Individuensymbole überein. Bei atomaren konventionellen Termen handelt es sich somit entweder um Variablen oder um Konstantensymbole.

Die Menge

$$ZT_{SIG_{KS}} = \{t \mid t \in TERM_{SIG_{KS}} \wedge t \notin (VAR \cup KON_{SIG_{KS}})\}$$
$$\text{und } ZT_{SIG_{KS}} = TERM_{SIG_{KS}} \setminus AT_{SIG_{KS}}$$

der *zusammengesetzten konventionellen Terme* umfasst alle konventionellen Terme, die aus der typgerechten Anwendung von Operationssymbolen auf atomare oder zusammengesetzte konventionelle Terme hervorgehen.

Die Konstruktion zusammengesetzter konventioneller Terme wird durch die induktive Definition von konventionellen Termen ermöglicht. Die *Induktionsbasis* bei der Definition der Menge $TERM_{SIG_{KS}}$ sind die Elemente aus der Menge $IND_{SIG_{KS}}$. Individuensymbole können in den Argumenten von Operationssymbolen verwendet werden, wodurch zusammengesetzte konventionelle Terme generiert werden. Darüber hinaus können in den Argumenten von Operationssymbolen auch zusammengesetzte konventionelle Terme vorkommen. Das Induktionsschema der oben aufgeführten Termdefinition erlaubt daher auch eine *verschachtelte* Termkonstruktion, indem im Argument von Operationssymbolen konventionelle zusammengesetzte Terme verwendet werden dürfen, die durch die Anwendung von Operationssymbolen auf (andere) Terme – entsprechend Regel (3.) – konstruiert wurden.

Zusammengesetzte Terme können sowohl mittels Eigennamen als auch mittels ihrer Konstruktionsvorschrift angesprochen werden. Ein zusammengesetzter konventioneller Term $t=O_i(t_1,...,t_m)$, der aus der Anwendung eines Operationssymbols $O_i \in OPS$ mit $typ_{OPS_{KS}}(O_i)=m$ und $m \in N$ auf ein m-Tupel $(t_1,...,t_m)$ von konventionellen Termen hervorgeht, kann somit sowohl als „der Term t“ als auch als „der Term $O_i(t_1,...,t_m)$“ angesprochen werden. Das gleiche gilt für die Terme $t_1,...,t_m$, die im Argument des Terms $O_i(t_1,...,t_m)$ vorkommen. Es kann beispielsweise sein, dass in dem m-Tupel $(t_1,...,t_m)$ von konventionellen Termen ein konventioneller zusammengesetzter Term t_x mit $t_x=O_y(t_{x_1},...,t_{x_y})$ vorkommt. Der „ausgefaltete“ Term t kann in diesem Fall sowohl als $t=O_i(t_1,...,O_y(t_{x_1},...,t_{x_y}),...,t_m)$ als auch als $t=O_i(t_1,...,t_x,...,t_n)$ angesprochen werden.

Das zweite Kriterium zur Klassifikation von konventionellen Termen betrifft ihren *Variablenanteil*. Die Menge $TERM_{SIG_{KS}}$ aller konventionellen Terme wird hinsichtlich ihres Variablenanteils in die zwei zueinander disjunkten Teilmengen $VT_{SIG_{KS}}$ und $GT_{SIG_{KS}}$ unterteilt:

$$TERM_{SIG_{KS}} = VT_{SIG_{KS}} \cup GT_{SIG_{KS}}$$
$$\text{mit } VT_{SIG_{KS}} \cap GT_{SIG_{KS}} = \varnothing.$$

Die Elemente der Menge VT_{KS} werden als *variable konventionelle Terme* bezeichnet. Bei variablen konventionellen Termen handelt es sich um solche konventionellen Terme, in denen mindestens eine Variable vorkommt. Die Menge $GT_{SIG_{KS}}$ umfasst hingegen alle Terme über einer konventionellen Signatur SIG_{KS}, in denen keine Variable vorkommt. Sie werden als *konventionelle Grundterme* bezeichnet.

Die Bestimmung des Variablenanteils konventioneller Terme erfolgt durch die Funktion

$$var_{T_{KS}}: TERM_{SIG_{KS}} \rightarrow pot(VAR).$$

Die Funktion $var_{T_{KS}}$ ordnet jedem Term $t \in TERM_{SIG_{KS}}$ über einer konventionellen Signatur SIG_{KS} eine Menge von Variablen aus der Menge VAR aller Variablen zu. Dabei gelten:[1)]

(1.) $var_{T_{KS}}(x_q) = \{x_q\}$
für alle $x_q \in VAR$,

(2.) $var_{T_{KS}}(O_i) = \varnothing$
für alle $O_i \in KON_{SIG_{KS}}$ und

(3.) $var_{T_{KS}}(O_i(t_1,...,t_m)) = var_{T_{KS}}(t_1) \cup ... \cup var_{T_{KS}}(t_m)$
für $O_i \in OPS$ und $t_1,...,t_m \in TERM_{SIG_{KS}}$ mit $typ_{OPS_{KS}}(O_i)=m$ und $m \in \mathbb{N}_+$.

Für atomare konventionelle Terme ist der Variablenanteil durch eine direkte Zuweisung bestimmt. Atomare konventionelle Terme gehen nämlich aus Individuensymbolen und somit entweder aus Variablen oder aus Konstantensymbolen hervor. Wenn es sich bei dem Individuensymbol um eine Variable handelt, entspricht der Variablenanteil des Terms der Variable selbst. Handelt es sich hingegen um ein Konstantensymbol, dann entspricht der Variablenanteil des Terms der leeren Menge $\varnothing$.

Für zusammengesetzte konventionelle Terme erfolgt ein *rekursiver* Aufruf der Funktion $var_{T_{KS}}$. Die Rekursion wird solange für die Terme in den Argumenten von Operationssymbolen weitergeführt, bis in dem jeweiligen Argument ein Individuensymbol vorkommt. Der Variablenanteil der Individuensymbole wird wiederum durch die Regeln (1.) und (2.) bestimmt.

Mit Hilfe der Funktion $var_{T_{KS}}$ können die Elemente der Mengen $VT_{SIG_{KS}}$ und $GT_{SIG_{KS}}$ präzise bestimmt werden. Die Menge

$$VT_{SIG_{KS}} = \{t \mid t \in TERM_{SIG_{KS}} \wedge var_{T_{KS}}(t) \neq \varnothing\}$$

aller variablen konventionellen Terme umfasst solche Terme, die mindestens eine Variable enthalten. Hierfür kommen insbesondere solche konventionellen Terme in Frage, die unmittelbar aus Variablen hervorgehen. Entsprechend ist die Menge VAR aller Variablen eine Teilmenge der Menge $VT_{SIG_{KS}}$ aller variablen Terme:

$$VAR \subseteq VT_{SIG_{KS}}.$$

Die Menge

$$GT_{SIG_{KS}} = \{t \mid var_{T_{KS}}(t) = \varnothing\}$$

aller konventionellen Grundterme umfasst solche Terme, die keine Variablen enthalten. Entsprechend ist die Menge $KON_{SIG_{KS}}$ aller Konstantensymbole eine Teilmenge der Menge $GT_{SIG_{KS}}$ aller Grundterme:

$$KON_{SIG_{KS}} \subseteq GT_{SIG_{KS}}.$$

1) Vgl. EBBINGHAUS ET AL. (1992), S. 28.

Den zwei Ausprägungsformen von Termen entsprechend können vier Arten von Termen über einer konventionellen Signatur SIG_{KS} unterschieden werden. In Tabelle 2 ist ein Überblick über die möglichen Ausprägungen von Termen gegeben.

	atomare Terme $AT_{SIG_{KS}}$	zusammengesetzte Terme $KT_{SIG_{KS}}$
Grundterme $GT_{SIG_{KS}}$	$t \in KON_{SIG_{KS}}$	$t = O_i(t_1,...,t_m)$ mit $var_{T_{KS}}(t)=\varnothing$
variable Terme $VT_{SIG_{KS}}$	$t \in VAR$	$t = O_i(t_1,...,t_m)$ mit $var_{T_{KS}}(t)\neq\varnothing$

Tabelle 2: Klassifikation konventioneller Terme

In den Zeilen der Tabelle sind die Ausprägungen des Kriteriums Variablenanteil gegeben. In den Spalten sind die Ausprägungen des Kriteriums Zusammengesetztheit gegeben. Für jeden Term $t \in TERM_{SIG_{KS}}$ kann mittels der Tabelle bestimmt werden, wie er sich nach Maßgabe der zwei genannten Kriterien klassifizieren lässt.

2.2.1.1.2.2.2 Konventionelle Formeln

Die zweite Teilmenge $FORM_{SIG_{KS}}$ der Menge $EXPR_{SIG_{KS}}$ aller Ausdrücke über einer konventionellen Signatur SIG_{KS} umfasst *konventionelle Formeln*. Die induktive Grammatik zur Konstruktion konventioneller Formeln lautet:[1)]

(1.) Es gilt $w,f \in FORM_{SIG_{KS}}$. w ist die immer gültige, tautologische Formel; f ist die immer ungültige, kontradiktorische Formel.

(2.) Wenn $R_j \in RS$, $typ_{RS_{KS}}(R_j)=n$ mit $n>0$ und $t_1,...,t_n \in TERM_{SIG_{KS}}$ gelten, dann gilt auch $R_j(t_1,...,t_n) \in FORM_{SIG_{KS}}$. Es handelt sich hierbei um eine *atomare konventionelle Formel*.

(3.) Wenn $F \in FORM_{SIG_{KS}}$ gilt, dann gilt auch $\neg F \in FORM_{SIG_{KS}}$. Es handelt sich hierbei um eine *zusammengesetzte konventionelle Formel*.

(4.) Wenn $F_1,F_2 \in FORM_{SIG_{KS}}$ gilt, dann gelten auch $F_1 \wedge F_2$, $F_1 \vee F_2$, $F_1 \underline{\vee} F_2$, $F_1 \rightarrow F_2$, $F_1 \leftrightarrow F_2 \in FORM_{SIG_{KS}}$. Es handelt sich hierbei um *zusammengesetzte konventionelle Formeln*.

(5.) Wenn $x_q \in VAR$ und $F \in FORM_{SIG_{KS}}$ gelten, dann gelten auch $\forall x_q{:}F$, $\exists x_q{:}F$, $\underline{\exists} x_q{:}F \in FORM_{SIG_{KS}}$. Es handelt sich hierbei um *zusammengesetzte konventionelle Formeln*.[2)]

1) Vgl. EBBINGHAUS ET AL. (1992), S. 20 („S-Ausdrücke“); ROTHMALER (1995), S. 30 f.; SCHÖNING (1992), S. 52; ZELEWSKI (1995), Bd. 4, S. 94.

2) Es wird bewusst durch die o.a. Formierungsregel der Freiraum gelassen, auch solche Formeln zu konstruieren, in denen die Quantoren sich über Variablen erstrecken, die nicht im Formelargument vorkommen. Beispielsweise ist der Ausdruck

Analog zur Menge $TERM_{SIG_{KS}}$ aller konventionellen Terme kann auch die Menge $FORM_{SIG_{KS}}$ aller konventionellen Formeln hinsichtlich der Kriterien *Zusammengesetztheit* und *Variablenanteil* ausdifferenziert werden. Darüber hinaus können konventionelle Formeln auch hinsichtlich ihrer *Offenheit* klassifiziert werden.

Die Zusammengesetztheit von konventionellen Formeln wird durch die induktive Definition der Menge $FORM_{SIG_{KS}}$ ermöglicht. Wie bei der Grammatik zur Konstruktion von konventionellen Termen auch, handelt es sich bei der Grammatik zur Konstruktion von konventionellen Formeln um eine *induktive* Definition. Die *Induktionsbasis* ist durch die Regeln (1.) und (2.) gegeben, nach der atomare konventionelle Formeln konstruiert werden.

Hinsichtlich der Zusammengesetztheit lassen sich *atomare konventionelle Formeln* aus der Menge $AF_{SIG_{KS}}$ und *zusammengesetzte konventionelle Formeln* aus der Menge $ZF_{SIG_{KS}}$ unterscheiden:

$$FORM_{SIG_{KS}} = AF_{SIG_{KS}} \cup ZF_{SIG_{KS}}$$
$$\text{mit } AF_{SIG_{KS}} \cap ZF_{SIG_{KS}} = \varnothing.$$

Die Zuordnung einer konventionellen Formel zu einer der beiden Mengen erfolgt auf der Basis der Teilformelfunktion

$$tf_{KS}: FORM_{SIG_{KS}} \rightarrow pot_{+}(FORM_{SIG_{KS}}).$$

Die Teilformelfunktion tf_{KS} ordnet jeder Formel $F \in FORM_{SIG_{KS}}$ über einer konventionellen Signatur SIG_{KS} eine Menge $FM \subseteq FORM_{SIG_{KS}}$ von Formeln über SIG_{KS} zu. Es handelt sich dabei stets um die Formelmenge FM, welche zum einen die Ausgangsformel F selbst und darüber hinaus alle Teilformeln von F enthält. Im Einzelnen gelten für alle Formeln $w, f, R_j(t_1,...,t_n), F, F_1, F_2 \in FORM_{SIG_{KS}}$:[1)]

(1.) $tf_{KS}(w) = \{w\}$,

(2.) $tf_{KS}(f) = \{f\}$,

(3.) $tf_{KS}(R_j(t_1,...,t_n)) = \{R_j(t_1,...,t_n)\}$,

(4.) $tf_{KS}(\neg F) = \{\neg F\} \cup tf_{KS}(F)$,

(5.) $tf_{KS}((F_1 \bullet F_2)) = \{(F_1 \bullet F_2)\} \cup tf_{KS}(F_1) \cup tf_{KS}(F_2)$ für $\bullet \in \{\wedge, \vee, \underline{\vee}, \rightarrow, \leftrightarrow\}$ und

(6.) $tf_{KS}(\bullet x{:}F) = \{\bullet x{:}F\} \cup tf_{KS}(F)$ für $\bullet \in \{\forall, \exists, \underline{\exists}\}$.

$$\forall x{:}\ R_j(y)$$

als konventionelle Formel zugelassen, wenn $typ_{RS_{KS}}(R_j)=1$ gilt. Hierdurch wird der „gewöhnliche" Fall von konventionellen Formeln, in denen sich die Quantifizierung über Variablen erstreckt, die im Formelargument vorkommen, auch umfasst. Zu einer derartigen Zulässigkeit der Konstruktion prädikatenlogischer Formeln vgl. EBBINGHAUS ET AL. (1992), S. 21.

1) Vgl. EBBINGHAUS ET AL. (1992), S. 28.

Die Elemente der Mengen $AF_{SIG_{KS}}$ und $ZF_{SIG_{KS}}$ können mit Hilfe der Teilformelfunktion tf_{KS} bestimmt werden. Es gelten für alle Formeln $F \in FORM_{SIG_{KS}}$ über einer konventionellen Signatur SIG_{KS}:

$$AF_{SIG_{KS}} = \{F \mid F \in FORM_{SIG_{KS}} \wedge tf_{KS}(F) = \{F\}\}$$
$$\text{und } ZF_{SIG_{KS}} = \{F \mid F \in FORM_{SIG_{KS}} \wedge tf_{KS}(F) \supset \{F\}\}.$$

Die Menge $AF_{SIG_{KS}}$ der atomaren konventionellen Formeln umfasst alle konventionellen Formeln F, deren Teilformelmenge $tf_{KS}(F)$ nur sie selbst enthält ($tf_{KS}(F)$=F). Da die Teilformelmenge $tf_{KS}(F)$ zu einer Formel F mindestens sich selbst als Element hat und für atomare konventionelle Formeln keine weiteren Formeln als Teilformeln erlaubt sind, beträgt die Mächtigkeit $|tf_{KS}(F)|$ der Teilformelmenge $tf_{KS}(F)$ einer atomaren konventionellen Formel F stets genau 1.

Atomare konventionelle Formeln gehen entweder unmittelbar aus den Formeln w und f oder mittelbar aus der Anwendung eines Relationssymbols R_j mit der Typisierung $typ_{RS_{KS}}(R_j)=n$ auf genau n konventionelle Terme hervor. Die konventionellen Terme, auf die das Relationssymbol R_j angewendet wird, werden bei der Anwendung von den Hilfszeichen „(“ und „)“ umfasst.

Zusammengesetzte Formeln gehen hingegen aus der Kombination konventioneller Formeln mit logischen Symbolen hervor. Die Zusammengesetztheit atomarer konventioneller Formel ist nur in Bezug auf ihren *Formel*aufbau definiert. Die Zusammengesetztheit der Terme im Argument atomarer konventioneller Formel hat keinen Einfluss auf die Zusammengesetztheit der Formeln. Demnach können atomare konventionelle Formeln in ihren Argumenten auch zusammengesetzte konventionelle Terme aufweisen. In diesem Fall wird von einer *tiefen Argumentstruktur* der atomaren konventionellen Formel gesprochen. Wenn das Formelargument einer atomaren Formel nur atomare konventionelle Terme aufweist, wird von einer *flachen Argumentstruktur* gesprochen.[1)]

Zusammengesetzte konventionelle Formeln sind solche konventionellen Formeln, die aus der Anwendung von mindestens einer der Regeln (4.) bis (6.) zur Konstruktion von Formeln über einer konventionellen Signatur SIG_{KS} hervorgegangen sind. Die Menge $ZF_{SIG_{KS}}$ aller zusammengesetzten konventionellen Formeln umfasst demnach alle Formeln F, deren Teilformelmenge $tf_{KS}(F)$ notwendigerweise sich selbst und hinreichenderweise mindestens eine weitere konventionelle Formel enthält. Da zusammengesetzte konventionelle Formeln mindestens sich selbst und eine andere konventionelle Formel als Teilformel haben, beträgt die Mächtigkeit $|tf(F)|$ einer zusammengesetzten konventionellen Formel F mindestens 2.

Der Variablenanteil einer konventionellen Formel wird durch ihr Bild entsprechend der Funktion

$$var_{F_{KS}}: FORM_{SIG_{KS}} \rightarrow pot(VAR)$$

1) Vgl. ZELEWSKI (1995), Bd. 4, S. 98.

bestimmt. Sie ordnet jeder konventionellen Formel F eine Teilmenge der Menge VAR aller Variablen zu. Es handelt sich hierbei um die Menge von Variablen, die in den Termen des Formelarguments enthalten sind. Dabei gelten:

(1.) $var_{F_{KS}}(w) = \varnothing$,

(2.) $var_{F_{KS}}(f) = \varnothing$,

(3.) $var_{F_{KS}}(R_j(t_1,...,t_n)) = var_{T_{KS}}(t_1) \cup ... \cup var_{T_{KS}}(t_n)$
für alle $R_j \in RS$ mit $typ_{RS_{KS}}(R_j)=n$ und $t_x \in TERM_{SIG_{KS}}$ mit $x \in \{1,...,n\}$,

(4.) $var_{F_{KS}}(\neg F) = var_{F_{KS}}(F)$
für alle konventionellen Formeln $F \in FORM_{SIG_{KS}}$,

(5.) $var_{F_{KS}}(F_1 \bullet F_2) = var_{F_{KS}}(F_1) \cup var_{F_{KS}}(F_2)$
für alle konventionellen Formeln $F_1,F_2 \in FORM_{SIG_{KS}}$ und $\bullet \in \{\wedge,\vee,\underline{\vee},\rightarrow,\leftrightarrow\}$ und

(6.) $var_{F_{KS}}(\bullet{:}F) = var_{F_{KS}}(F) \cup \{x_q\}$
für alle konventionellen Formeln $F \in FORM_{SIG_{KS}}$ und $\bullet \in \{\forall x_q, \exists x_q, \underline{\exists} x_q\}$.

Bei der Funktion $var_{F_{KS}}$ zur Bestimmung des Variablenanteils von konventionellen Formeln handelt es sich – wie bei der Funktion $var_{T_{KS}}$ zur Bestimmung des Variablenanteils von konventionellen Termen auch – um eine *rekursive* Funktion. Während bei $var_{T_{KS}}$ die Termination des rekursiven Funktionsaufrufs durch atomare Terme gewährleistet ist, wird dies bei $var_{F_{KS}}$ durch atomare Formeln übernommen. Der Variablenanteil von der konventionellen Formeln w und f ist stets $\varnothing$. Bei atomaren Formeln, die aus der typgerechten Anwendung eines Relationssymbols R_j mit $typ_{RS_{KS}}(R_j)=n$ auf ein Termtupel $(t_1,...,t_n)$ hervorgehen, terminiert die Funktion $var_{F_{KS}}$ dann, wenn die Funktion $var_{T_{KS}}$ terminiert. Der Variablenanteil $var_{F_{KS}}(R_j(t_1,...,t_n))$ einer atomaren Formel $R_j(t_1,...,t_n)$ stimmt mit der Vereinigung aller Variablenanteile der konventionellen Terme $t_1,...,t_n$ überein, die im Argument der Formel vorkommen.

Bei der Bestimmung des Variablenanteils zusammengesetzter Formeln kann es zu zwei voneinander unabhängigen Rekursionen kommen. Zum ersten handelt es sich um den rekursiven Aufruf der Funktion $var_{F_{KS}}$, um den Variablenanteil der Teilformeln von zusammengesetzten konventionellen Formeln bestimmen zu können. Zum zweiten handelt es sich um den rekursiven Aufruf der Funktion $var_{T_{KS}}$, wenn es sich bei den konventionellen Termen, die in den Argumenten der Relationssymbole verwendet werden, um zusammengesetzte konventionelle Terme handelt. Für beide Funktionen ist ihre Termination durch ihre jeweiligen Rekursionsbasen gewährleistet.

Konventionelle Formeln werden hinsichtlich ihres Variablenanteils in *konventionelle Grundformeln* und *variable konventionelle Formeln* unterschieden. Für diese Unterscheidung wird die Menge $FORM_{SIG_{KS}}$ aller Formen über einer konventionellen Signatur SIG_{KS} in die Menge $GF_{SIG_{KS}}$ aller konventionellen Grundformeln und die Menge $VF_{SIG_{KS}}$ aller variablen konventionellen Formeln unterteilt:

$$FORM_{SIG_{KS}} = GF_{SIG_{KS}} \cup VF_{SIG_{KS}} \text{ mit}$$
$$GF_{SIG_{KS}} \cap VF_{SIG_{KS}} = \varnothing.$$

Die Elemente der Menge $GF_{SIG_{KS}}$ weisen in den Argumenten ihrer atomaren Teilformeln höchstens[1)] konventionelle Grundterme aus der Menge $GT_{SIG_{KS}}$ auf. Der Variablenanteil konventioneller Grundformeln entspricht somit der leeren Menge $\varnothing$. Ansonsten werden konventionelle Formeln zur Menge $VF_{SIG_{KS}}$ aller variablen konventionellen Formeln gezählt. Der Variablenanteil der konventionellen Formeln aus der Menge $VF_{SIG_{KS}}$ ist somit ungleich der leeren Menge $\varnothing$. Mit Hilfe der Funktion $var_{F_{KS}}$ können die Elemente der Mengen GF_{KS} und VF_{KS} wie folgt bestimmt werden:

$$GF_{SIG_{KS}} = \{F \mid F \in FORM_{SIG_{KS}} \wedge var_{F_{KS}}(F) = \varnothing\}$$
$$\text{und } VF_{SIG_{KS}} = \{F \mid F \in FORM_{SIG_{KS}} \wedge var_{F_{KS}}(F) \supset \varnothing\}.$$

Hinsichtlich ihrer Offenheit können konventionelle Formen in *offene* und *geschlossene konventionelle Formeln* unterschieden werden. Dafür wird die Menge $FORM_{SIG_{KS}}$ aller Formeln über einer konventionellen Signatur SIG_{KS} in die Menge $OF_{SIG_{KS}}$ aller offenen konventionellen Formeln und die Menge $CF_{SIG_{KS}}$ aller geschlossenen konventionellen Formeln unterteilt:

$$FORM_{SIG_{KS}} = OF_{SIG_{KS}} \cup CF_{SIG_{KS}}$$
$$\text{mit } OF_{SIG_{KS}} \cap CF_{SIG_{KS}} = \varnothing.$$

Die Menge $OF_{SIG_{KS}}$ aller offenen konventionellen Formeln enthält konventionelle Formeln, die in ihren Argumenten mindestens eine *freie Variable* enthalten. Eine Variable x_q kommt im Argument einer konventionellen Formel F genau dann frei vor, wenn sie in F durch keinen Quantor gebunden wird. Da in offenen konventionellen Formeln mindestens eine freie Variable vorkommen muss, handelt es sich bei ihnen grundsätzlich um variable konventionelle Formeln.

Die Menge $CF_{SIG_{KS}}$ aller geschlossenen konventionellen Formeln enthält konventionelle Formeln, die in ihren Argumenten keine freien Variablen enthalten. Die Variablen in den Argumenten geschlossener konventioneller Formeln werden als *gebundene Variablen* bezeichnet. Wenn eine Formel keine Variablen enthält, wird sie definitorisch zu den geschlossenen Formeln gerechnet. Bei konventionellen Grundformeln handelt es sich daher grundsätzlich um geschlossene konventionelle Formeln.

Die Bestimmung des Anteils an freien Variablen, die in einer konventionellen Formel vorkommen, erfolgt über die Funktion

$$fvar_{KS}: FORM_{SIG_{KS}} \rightarrow pot(VAR).$$

Sie ordnet jeder Formel $F \in FORM_{SIG_{KS}}$ über einer konventionellen Signatur SIG_{KS} eine Teilmenge der Menge VAR aller Variablen zu. Dabei gelten:[2)]

(1.) $fvar_{KS}(w) = \varnothing$,

(2.) $fvar_{KS}(f) = \varnothing$,

---

1) Für den Grenzfall, dass es sich bei der atomaren Teilformel um eine der beiden Formeln w oder f handelt, weist die Teilformel keinen Term in ihrem Argument auf.

2) Vgl. EBBINGHAUS ET AL. (1992), S. 30; EHRIG ET AL. (1999), S. 316; GALLIER (1986), S. 154.

(3.) $fvar_{KS}(R_j(t_1,...,t_n)) = var_{T_{KS}}(t_1) \cup ... \cup var_{T_{KS}}(t_n)$
für alle $R_j \in RS$ mit $typ_{RS_{KS}}(R_j)=n$ und $t_1,...,t_n \in TERM_{SIG_{KS}}$,

(4.) $fvar_{KS}(\neg F) = fvar_{KS}(F)$
für alle $F \in FORM_{SIG_{KS}}$,

(5.) $fvar_{KS}(F_1 \bullet F_2) = fvar_{KS}(F_1) \cup fvar_{KS}(F_2)$
für alle $F_1,F_2 \in FORM_{SIG_{KS}}$ und $\bullet \in \{\wedge,\vee,\underline{\vee},\rightarrow,\leftrightarrow\}$,

(6.) $fvar_{KS}(\bullet F) = fvar_{KS}(F) \setminus \{x_q\}$
für alle $F \in FORM_{SIG_{KS}}$ und $\bullet \in \{\forall x_q,\exists x_q:,\underline{\exists} x_q:\}$.

Eine Variable $x_q \in VAR$ kommt in einer konventionellen Formel $F \in FORM_{SIG_{KS}}$ genau dann *frei* vor, wenn sie in F nicht durch einen Quantor gebunden ist. Ansonsten kommt x_q in F in *gebundener* Form vor. Wenn es sich bei $F \in FORM_{SIG_{KS}}$ um eine atomare konventionelle Formel handelt, dann sind alle Variablen im Argument von F grundsätzlich frei, da in atomaren konventionellen Formeln keine Quantoren vorkommen können.

Das Bild einer konventionellen Formel $F \in FORM_{SIG_{KS}}$ entsprechend der Funktion $fvar_{KS}$ ist die Menge von Variablen, die in F frei vorkommen. Mit Hilfe der Funktion $fvar_{KS}$ lassen sich die Menge $OF_{SIG_{KS}}$ aller offenen Formeln und die Menge $CF_{SIG_{KS}}$ aller geschlossenen Formeln über einer konventionellen Signatur SIG_{KS} bestimmen. Es gelten für alle konventionellen Formeln $F \in FORM_{SIG_{KS}}$:

$$OF_{SIG_{KS}} = \{F \mid F \in FORM_{SIG_{KS}} \wedge fvar_{KS}(F) \supset \varnothing\}$$
$$\text{und } CF_{SIG_{KS}} = \{F \mid F \in FORM_{SIG_{KS}} \wedge fvar_{KS}(F) = \varnothing\}.$$

In konventionellen Grundformeln kommen grundsätzlich keine Variablen vor. Somit können in konventionellen Grundformeln auch keine freien Variablen vorkommen. Entsprechend ist die Menge $GF_{SIG_{KS}}$ aller konventionellen Grundformeln eine Teilmenge der Menge $CF_{SIG_{KS}}$ aller geschlossenen konventionellen Formeln:

$$GF_{SIG_{KS}} \subseteq CF_{SIG_{KS}}.$$

2.2.1.1.3 Semantische Aspekte der konventionellen Prädikatenlogik

2.2.1.1.3.1 SIG_{KS}-Strukturen

Sowohl Terme als auch Formeln über einer konventionellen Signatur SIG_{KS} wurden bislang als Ausdrücke vorgestellt, deren *Bedeutung* zunächst ungeklärt ist. Obwohl die Ausdrücke über einer konventionellen Signatur SIG_{KS} bislang „bedeutungslos“ sind, kann auch schon die die Festlegung der Grammatik zu ihrer Konstruktion ohne eine Erweiterung für unterschiedliche Zwecke nützlich sein. Das Signaturkonzept kann z.B. für Software zur Überprüfung prädikatenlogisch konstruierter Aussagen hinsichtlich ihrer *Korrektheit* ausreichend sein. Eine ähnliche Nützlichkeit ist beispielsweise bei der

Backus-Naur-Notation[1)] für formale Sprachen zu beobachten. Dennoch kann mit dem Signaturkonzept allein das Potenzial der Prädikatenlogik noch nicht annähernd ausgeschöpft werden.

Auch wenn die natürlichsprachliche Bezeichnung der Komponenten von Signaturen, mit denen Ausdrücke konstruiert werden, oft *eine* bestimmte Interpretation nahe legt, handelt es sich in diesen Fällen um keine präzise Festlegung der Semantik. Natürlichsprachliche Interpretationen von formal- oder wiederum natürlichsprachlichen Konstrukten können nämlich mit *Mehrdeutigkeiten* behaftet sein.

Bei der Semantik für prädikatenlogische Ausdrücke, die in der vorliegenden Arbeit vorgestellt wird, handelt es sich um eine *formale Semantik*. Den Ausdrücken, die mit Hilfe des konventionellen Alphabets $ALPH_{KS}$ und den Regeln zur Konstruktion von Termen und Formeln aufgebaut werden können, werden innerhalb einer formalen Semantik *formale Objekte* bzw. *formale Wahrheitswerte* zugeordnet. Dazu ist es notwendig, die Komponenten einer konventionellen Signatur SIG_{KS} *extensional zu interpretieren*. Die extensionale Interpretation erfolgt, indem Termen formale Objekte und Formeln formale Wahrheitswerte zugeordnet werden.

Eine formale Semantik, die Termen formale Objekte und Formeln formale Wahrheitswerte zuordnet, basiert auf dem Prinzip der modelltheoretischen Semantik nach TARSKI.[2)] Den Grundbaustein einer modelltheoretischen Semantik bildet die Abbildung von Termen auf formale Objekte. Der Abbildungsbereich dieser formalen Semantik ist eine so genannte *SIG_{KS}-Struktur*.[3)]

Eine SIG_{KS}-Struktur $A_{SIG_{KS}}$ zu einer konventionellen Signatur SIG_{KS} ist definiert als:

$$A_{SIG_{KS}}=(OB,OPF,RF,IF_{KS}).$$

Die Komponenten einer SIG_{KS}-Struktur $A_{SIG_{KS}}$ zu einer konventionellen Signatur SIG_{KS} sind:

(1.) eine *Objektmenge* $OB=\{ob_1,...,ob_U\}$ mit $u=1,...,U$ und $U\in\mathbb{N}_+$,

(2.) eine Familie $OPF=(o_1,...,o_I)$ von *Operationen* o_i mit $i=1,...,I$ und $I\in\mathbb{N}$ und

(3.) eine Familie $RF=(r_1,...,r_J)$von *Relationen* r_j mit $j=1,...,J$ und $J\in\mathbb{N}$ und

(4.) eine Familie $IF_{KS}=(I_{OPS},I_{RS})$ von *Interpretationsfunktionen*.

Die Objektmenge OB wird synonym auch als *Träger*- oder *Individuenmenge* bezeichnet und konstituiert das *prädikatenlogische Universum*.[4)] Die Elemente der Objektmenge OB sind formale Objekte, die als *Individuen* bezeichnet werden. Individuen sind die

---

1) Vgl. EIRUND ET AL. (2000), S. 51 ff.

2) Vgl. SCHMID/KINDSMÜLLER (1996), S. 278 f.

3) Logische Strukturen werden auch verkürzt als Strukturen angesprochen werden, wenn aus dem Argumentationskontext ersichtlich ist, dass es sich um logische Strukturen handelt.

4) Vgl. ROTHMALER (1995), S. 18; ZELEWSKI (1995), Bd. 4, S. 114.

Grundbausteine der modelltheoretischen Semantik für eine prädikatenlogische Sprache. Sowohl die Operationen als auch die Relationen aus den Familien OPF bzw. RF werden nämlich als Teilmengen von kartesischen Produkten der Objektmenge OB mit sich selbst definiert.

Die Verknüpfung zwischen syntaktischen Ausdrücken über einer konventionellen Signatur SIG_{KS} und semantischen Komponenten einer SIG_{KS}-Struktur erfolgt durch die Mitglieder der Familie $IF_{KS}=(I_{OPS},I_{RS})$ von extensionalen Interpretationsfunktionen. Es handelt sich hierbei um Funktionen, die die Zuweisung von Extensionen zu Operations- und Relationssymbolen aus der konventionellen Signatur SIG_{KS} erlauben. Entsprechend wird die Anwendung einer Interpretationsfunktion I_{OPS} oder I_{RS} auf ein deskriptives Symbol als dessen *extensionale Interpretation* oder kurz *Extension* bezeichnet.

Bei den Interpretationsfunktionen I_{OPS} und I_{RS} handelt es sich um linkstotale, bijektive[1)] Funktionen, mit denen Konstrukten aus einer konventionellen Signatur SIG_{KS} eineindeutig Konstrukte aus einer SIG_{KS}-Struktur $A_{SIG_{KS}}$ zugeordnet werden können. Hierbei werden *alle* Operations- und Relationssymbole aus einer konventionellen Signatur SIG_{KS} durch Operationen bzw. Relationen aus einer SIG_{KS}-Struktur $A_{SIG_{KS}}$ extensional interpretiert. Zudem sind *alle* Operationen und Relationen aus einer SIG_{KS}-Struktur $A_{SIG_{KS}}$ extensionale Interpretationen von jeweils genau einem Operations- bzw. Relationssymbol aus der konventionellen Signatur SIG_{KS}.[2)] Insofern ist jede SIG_{KS}-Struktur $A_{SIG_{KS}}$ zu einer konventionellen Signatur SIG_{KS} „passend“ zu SIG_{KS} aufgebaut.

---

1) Da es sich bei den Interpretationsfunktionen um *bijektive* Funktionen handelt, werden sie bei der formalen Definition von Strukturen zu konventionellen Signaturen oftmals ausgelassen; vgl. z.B. ROTHMALER (1995), S. 18 ff.; EHRIG ET AL. (1999), S. 312 (für die sortierte Prädikatenlogik). Eine SIG_{KS}-Struktur $A_{SIG_{KS}}$ zu einer konventionellen Signatur SIG_{KS} wird in diesen Fällen *implizit* als „passend“ zu SIG_{KS} aufgefasst. Alternativ werden in einigen Ansätzen lediglich die Interpretationsfunktionen aus I_{KS} aufgeführt, und auf die Angaben der Mengen OPF und RF wird verzichtet; vgl. z.B. EBBINGHAUS ET AL. (1992), S. 35; SCHÖNING (1992), S. 54. In der vorliegenden Arbeit werden grundsätzlich alle Komponenten einer SIG_{KS}-Struktur $A_{SIG_{KS}}$ explizit aufgeführt.

2) Die eineindeutige Zuordnung von Operationen und Relationen zu Operations- bzw. Relationssymbolen ist eine Basisentscheidung, die der vorliegenden Arbeit zugrunde liegt. Üblicherweise wird die Eineindeutigkeit der extensionalen Interpretationsfunktionen nicht zwingend eingefordert. Dadurch wird die Möglichkeit frei gelassen, solche Operationen und Relationen zu spezifizieren, durch die mehrere Symbole extensional interpretiert werden. Beispielsweise könnten zwei Relationssymbole R_1 und R_2 durch die gleiche Relation r_j extensional interpretiert werden, wenn die extensionale Interpretationsfunktion I_{RS} nicht linkseindeutig wäre. Die beiden Relationssymbole könnten in diesem Fall als zueinander „synonyme“ Symbole aufgefasst werden.

Das sprachliche Phänomen der *Synonymie* wird in der vorliegenden Arbeit auf eine metasprachliche Ebene angesiedelt. Zueinander synonyme Zeichenketten werden in Abschnitt 3.1.3.2.2.3 als metasprachliche Bezeichner für objektsprachliche Symbole eingeführt. Darüber hinaus wird auch die *Homonymie* von Zeichenketten in 3.1.3.2.2.3 vorgestellt. Es handelt sich hierbei um solche metasprachlichen Zeichenketten, mit denen mindestens zwei objektsprachliche Symbole bezeichnet werden.

Mit der Interpretationsfunktion

$$I_{OPS}: OPS \rightarrow OPF$$

wird jedes Operationssymbol $O_i \in OPS$ durch genau eine Operation $o_i \in OPF$ extensional interpretiert. Dabei ordnet die Interpretationsfunktion I_{OPS} jedem Operationssymbol O_i eine $typ_{OPS_{KS}}(O_i)$-stellige Operation $o_i \in OPF$ über OB zu. Jede Operation $o_i \in OPF$ zu einem Operationssymbol O_i ist eine rechtseindeutige Relation – also eine *Abbildung* oder *Funktion* – der Form

$$o_i: OB^m \rightarrow OB$$
$$\text{mit } m = typ_{OPS_{KS}}(O_i).$$

Für Operationssymbole, die in der Form $typ_{OPS_{KS}}(O_i)=0$ typisiert sind, entspricht das Bild der Interpretationsfunktion I_{OPS} einer Operation $o_i \in OPF$ mit der Operationsvorschrift

$$o_i: \rightarrow OB.$$

Operationen dieser Art werden als *Konstanten* bezeichnet. Es handelt sich in diesem Fall um Operationen, deren Anwendung auf das leere Argument das Individuum $I_{OPS}(O_i)=o_i()=ob_u$ aus dem prädikatenlogischen Universum OB hervorbringt.

Mit der Interpretationsfunktion

$$I_{RS}: RS \rightarrow RF$$

wird jedes Relationssymbol $R_j \in RS$ durch eine Relation $r_j \in RF$ der Form

$$r_j \subseteq OB^m$$
$$\text{mit } m = typ_{RS_{KS}}(R_j).$$

extensional interpretiert. Sie ordnet jedem Relationssymbol $R_j \in RS$ eine $typ_{RS_{KS}}(R_j)$-stellige Relation $r_j \in RF$ über OB zu.

Die Extension eines Relationssymbols R_j ist die Menge aller n-Tupel $(ob_1,...,ob_n)$ aus formalen Objekten ob_x mit $x=1,...,n$ und $ob_x \in OB$, die die Relation $r_j(ob_1,...,ob_n)$ in der SIG_{KS}-Struktur $A_{SIG_{KS}}$ zu einer konventionellen Signatur SIG_{KS} erfüllen. Durch diesen Ansatz wird eine extensionale Semantik aller konventionellen Formeln aus der Menge $FORM_{SIG_{KS}}$ fundiert.[1] Diese Semantik weist jedem Relationssymbol R_j eine Menge formal definierter Konstrukte aus der SIG_{KS}-Struktur $A_{SIG_{KS}}$ als extensionale Interpretation der konventionellen Signatur SIG_{KS} zu. *Intensionale* Semantiken für Relationssymbole werden im Rahmen der konventionellen Prädikatenlogik nicht berücksichtigt.[2]

1) Vgl. ZELEWSKI (1995), Bd. 4, S. 115.

2) Die Beschränkung auf extensionale Semantiken gilt allerdings nur bei der Interpretation von Konstrukten aus konventionellen und sortierten Signaturen. Sortierte Signaturen werden in den nächsten Abschnitten vorgestellt. Später wird auch ein Ansatz verfolgt, Konstrukten aus ontologischen Signaturen mittels natürlichsprachlicher Definitionen eine informale, intensionale Semantik zukommen zu lassen.

Die Bilder der Interpretationsfunktionen I_{OPS} und I_{RS} werden in der vorliegenden Arbeit durch die verwendete Schreibweise verdeutlicht. Jede Interpretation $I_{OPS}(O_i)$ und $I_{RS}(R_j)$ eines Operations- bzw. Relationssymbols entspricht demnach einer Operation $o_i \in OPF$ bzw. einer Relation $r_j \in RF$. Um eine zweifache Redundanz[1)] in der Schreibweise zu vermeiden, wird daher im Folgenden auf die Angabe der Interpretationsfunktionen bei fehlendem Bedarf zur expliziten Ausführung verzichtet.

SIG_{KS}-Strukturen werden in Anlehnung an die Schreibweise für konventionelle Signaturen entsprechend folgendem Schema notiert:[2)]

$\underline{\underline{A}}_{\underline{SIG_{KS}}}$

$OB=\{ob_1,...,ob_U\}$

$OPF \quad =(o_1,...,o_I)$

$\qquad o_1: OB^{l(o1)} \rightarrow OB$

$\qquad ...$

$\qquad o_I: OB^{l(oI)} \rightarrow OB$

$RF \quad =(r_1,...,r_J)$

$\qquad r_1 \subseteq OB^{l(r1)}$

$\qquad ...$

$\qquad r_J \subseteq OB^{l(rJ)}$.

Jede SIG_{KS}-Struktur $A_{SIG_{KS}}$ zu einer konventionellen Signatur SIG_{KS} liefert *eine* mögliche extensionale Interpretation der deskriptiven Symbole aus der konventionellen Sig-

---

1) Die zweifache Redundanz resultiert daraus, dass die Interpretation von Operations- und Relationssymbolen zum einen durch die Großschrift der Symbole und zum anderen durch die Kleinschrift der Operationen bzw. Relationen verdeutlicht wird. So wird das Relationssymbol P durch die Relation p und das Relationssymbol Q durch die Relation q extensional interpretiert. Von dieser Schreibweise wird insbesondere in der Fallstudie im Anhang Gebrauch gemacht. Diese Vorgehensweise birgt die erste Redundanz gegenüber der Anwendung der Interpretationsfunktionen I_{OPS} bzw. I_{RS}. Die zweite Redundanz resultiert aus den Indizes, die Operations- und Relationssymbolen angeheftet werden. Jedes Operations- oder Relationssymbol O_i bzw. R_j wird nämlich durch eine indexgleiche Operation o_i bzw. Relation r_j interpretiert.

2) In der SIG_{KS}-Struktur wird als Index zu Objekten aus Objekttupeln u.a. die Bezeichnung $l(o_1)$ verwendet. l kann als eine Funktion aufgefasst werden, die das Konstrukt (z.B. o_1), das dem Objekttupel vorangestellt ist, auf eine natürliche Zahl $l(o_1)$ abbildet, um die numerische Stelligkeit des jeweils verwendeten Konstrukts (Operation oder Relation) anzugeben. Im Fall der konventionellen Prädikatenlogik stimmt beispielsweise für eine Operation o_i der Wert $l(o_i)$ mit dem Typ $typ_{OPS_{KS}}(O_i)$ des Operationssymbols O_i mit $I_{OPS}(O_i)=o_i$ überein. Im späteren Fall der sortierten Prädikatenlogik wird die Typisierung jedoch nicht numerisch, sondern inhaltlich angegeben. Die inhaltliche Typisierung umfasst zwar implizit die numerische Typisierung, stimmt jedoch nicht mit dieser überein. Insofern unterscheiden sich in diesen Fällen $l(o_i)$ und $typ_{OPS_{SS}}(O_i)$ voneinander, wobei $typ_{OPS_{SS}}$ die Typisierungsfunktion für Operationssymbole im Rahmen der sortierten Prädikatenlogik ist. Sie wird in Abschnitt 2.2.1.2.2.1 näher vorgestellt.

natur SIG_{KS}. Es können grundsätzlich mehrere alternative Interpretationen der gleichen konventionellen Signatur SIG_{KS} in Frage kommen. Die Menge $A(SIG_{KS})$ umfasst *alle* Strukturen zu einer konventionellen Signatur SIG_{KS}.

Jedes Element $A_{SIG_{KS}}$ aus der Menge $A(SIG_{KS})$ umfasst eine Objektmenge OB, eine Familie OPF von Operationen, eine Familie RF von Relationen und eine Familie IF_{KS} von Interpretationsfunktionen. Da jede SIG_{KS}-Stuktur $A_{SIG_{KS}} \in A(SIG_{KS})$ auch die zwei bijektiven Interpretationsfunktionen I_{OPS} und I_{RS} umfasst, sind alle Strukturen aus $A(SIG_{KS})$ weiterhin „passend“ zu SIG_{KS} konstruiert.

2.2.1.1.3.2 Auswertung von konventionellen Ausdrücken

2.2.1.1.3.2.1 Auswertung von konventionellen Termen

Durch ihre *Auswertung* werden „bedeutungslose“ Ausdrücke über einer konventionellen Signatur SIG_{KS} um eine formale Semantik angereichert. Für eine solche Auswertung wird eine SIG_{KS}-Struktur $A_{SIG_{KS}}$ vorausgesetzt, wie sie im vorherigen Abschnitt vorgestellt wurde. Im Rahmen der Auswertung wird eine konventionelle Signatur SIG_{KS} durch eine SIG_{KS}-Struktur $A_{SIG_{KS}}$ interpretiert.

Entsprechend der vorherigen Unterteilung der Menge $EXPR_{SIG_{KS}}$ aller konventionellen Ausdrücke in die Menge $TERM_{SIG_{KS}}$ aller konventionellen Terme und die Menge $FORM_{SIG_{KS}}$ aller konventionellen Formeln werden auch zwei Arten von Auswertungen benötigt. Für beide Arten von Ausdrücken werden eigene Auswertungsverfahren definiert. Jedes Auswertungsverfahren umfasst die Menge alle Regeln, um Ausdrücke der entsprechenden Art zu interpretieren. Das erste Auswertungsverfahren bezieht sich auf die Zuordnung *formaler Objekte* aus einer SIG_{KS}-Struktur $A_{SIG_{KS}}$ zu konventionellen Termen. Das zweite Auswertungsverfahren bezieht sich auf die Zuordnung von *Wahrheitswerten* zu konventionellen Formeln.

Für die Auswertung von konventionellen Termen wird die *Termauswertungsfunktion*

$$IT_{KS}: TERM_{SIG_{KS}} \rightarrow OB$$

verwendet. Die Termauswertungsfunktion IT_{KS} ordnet jedem konventionellen Term $t \in TERM_{SIG_{KS}}$ ein formales Objekt $IT_{KS}(t)=ob_u$ aus der Objektmenge OB einer SIG_{KS}-Struktur $A_{SIG_{KS}}$ zu.[1)]

1) Bei einer vollständig formalen Argumentation müsste die Termauswertungsfunktion IT_{KS} als

$$IT_{KS}: TERM_{SIG_{KS}} \times A(SIG_{KS}) \rightarrow OB$$

Bei der Grammatik zur Konstruktion von konventionellen Termen wurden hinsichtlich ihrer Variabilität zwei Arten von konventionellen Termen unterschieden. Die erste Termmenge $GT_{SIG_{KS}}$ umfasst konventionelle Grundterme, in denen keine Variablen vorkommen dürfen. Die zweite Termmenge $VT_{SIG_{KS}}$ umfasste variable konventionelle Terme, in denen mindestens eine Variable vorkommt. Während für konventionelle Grundterme die Auswertung unmittelbar mittels der Termauswertungsfunktion IT_{KS} erfolgen kann, muss für variable konventionelle Terme zunächst ihr Variablenanteil extensional interpretiert werden. Dies geschieht, in dem eine Funktion eingeführt wird, die die *Belegung* von Variablen mit formalen Objekten aus einer SIG_{KS}-Struktur $A_{SIG_{KS}}$ erlaubt.

Die extensionale Interpretation von Variablen erfolgt durch die (konventionelle) *Variablenbelegungsfunktion*[1)]

$$bel_{KS}: VAR \rightarrow OB.$$

Die Variablenbelegung bel_{KS} ordnet jeder Variablen x_q ein formales Objekt ob_u zu. Das Bild $bel_{KS}(x_q)=ob_u$ einer Variablen x_q entsprechend der Variablenbelegung bel_{KS} wird als *Wert* der Variablen x_q entsprechend bel_{KS} bezeichnet.

Die Variablenbelegung bel_{KS} hat den Charakter eines Parameters für die Interpretation von Ausdrücken über einer Signatur SIG_{KS}. Je nachdem, welche Variablenbelegung verwendet wird, können nämlich – im Fall von variablen Ausdrücken – unterschiedliche Auswertungen der gleichen Ausdrücke bewirkt werden. Sowohl bei variablen Termen als auch bei variablen Formeln über einer konventionellen Signatur SIG_{KS} können unterschiedliche Auswertungen vorliegen, wenn auch unterschiedliche Variablenbelegungen vorgenommen werden. Im Fall variabler Formeln gilt dies allerdings nur für freie Variablen.

Für die Auswertung von variablen Formeln, in denen mindestens eine Variable durch einen Quantor gebunden ist, wird eine bedingte Variablenbelegung benötigt.

Die *bedingte konventionelle Variablenbelegung*

---

definiert werden. In diesem Fall würde jedem Tupel $(t,A_{SIG_{KS}})$, bestehend aus einem Term $t \in TERM_{SIG_{KS}}$ über der konventionellen Signatur SIG_{KS} und einer SIG_{KS}-Struktur $A_{SIG_{KS}} \in A(SIG_{KS})$, ein formales Objekt $IT_{KS}(t,A_{SIG_{KS}})=ob_u$ aus der Trägermenge OB von $A_{SIG_{KS}}$ zugewiesen werden. Ebenso müsste die Variablenbelegungsfunktion bel_{KS} explizit benannt werden, die weiter unten vorgestellt wird. Der einfacheren Diktion halber wird hier allerdings auf beides verzichtet. Welche SIG_{KS}-Struktur $A_{SIG_{KS}}$ und welche Variablenbelegungsfunktion bel_{KS} jeweils gemeint sind, wird in den natürlichsprachlichen Erläuterungen bei Bedarf hervorgehoben.

1) Der Zusatz „konventionelle“ wird im Folgenden nur dann aufgeführt, wenn eine Abgrenzung zu sortierten oder ontologischen Variablenbelegungsfunktionen nötig ist. Darüber hinaus wird die verkürzte Bezeichnung „Variablenbelegung“ zugelassen.

$$bel_{KS}[x^*/ob_u](x)=\begin{cases} bel_{KS}(x) & \text{wenn } x \neq x^* \\ ob_u & \text{wenn } x = x^* \end{cases}$$

weist jeder Variablen x ein formales Objekt entsprechend der Variablenbelegung bel_{KS} zu, wenn sich die Variable x von einer näher spezifizierten Variable x* unterscheidet. Ansonsten wird der Variable x=x* das formale Objekt ob_u zugeordnet, für das $bel_{KS}[x^*,ob_u]$ definiert ist.

Bei der bedingten Variablenbelegung wird der nicht-deterministische Charakter von Variablenbelegungen teilweise aufgehoben. Die Nicht-Determiniertheit gilt dann nämlich nur für alle Variablen außer der spezifizierten Variable x*. Die Belegung von x* geht wiederum als Bedingung in die gesamte Variablenbelegung ein. Eine derartige Bedingung wird für die Auswertung von Formeln mit Quantoren benötigt, um die quantifizierten Variablen – entsprechend dem jeweils verwendeten Quantor – zu belegen. Wenn beispielsweise eine geschlossene konventionelle Formel $\forall x$: R(x) ausgewertet werden soll, wird eine bedingte Belegung der Variable x mit *allen* formalen Objekten aus dem Universum unterstellt.

Die Auswertung von konventionellen Termen erfolgt nach folgendem Rekursionsschema:

(1.) $IT_{KS}(x_q)=bel_{KS}(x_q)=ob_u$ mit $ob_u \in OB$
für jede Variable $x_q \in VAR$,

(2.) $IT_{KS}(O_i)=I_{OPS}(O_i)=o_i()$ mit $o_i()=ob_u$ und $ob_u \in OB$,
für jedes Konstantensymbol $O_i \in OPS$ mit $typ_{OPS_{KS}}(O_i)=0$ und

(3.) $IT_{KS}(O_i(t_1,...,t_n))=o_i(IT_{KS}(t_1),...,IT_{KS}(t_n))$
für jedes Operationssymbol $O_i \in OPS$ mit $typ_{OPS_{KS}}(O_i)=n$, $n \in \mathbb{N}_+$ und
$t_1,...,t_n \in TERM_{SIG_{KS}}$.

Mit Regel (1.) wird festgelegt, dass die Auswertung jedes konventionellen Terms t, der die Variable $x_q \in VAR$ ist, mit der Variablenbelegung $bel_{KS}(x_q)$ von x_q übereinstimmt. Somit ist die Auswertung von variablen konventionellen Termen von der jeweiligen Variablenbelegung bel_{KS} abhängig. Je nachdem, welche Variablenbelegung bel_{KS} vorausgesetzt wird, variiert die Auswertung von variablen Termen. Für Grundterme ist hingegen ihre Auswertung von der Interpretationsfunktion I_{OPS}[1)] abhängig.

Mit Regel (2.) wird festegelegt, dass die Auswertung $IT_{KS}(O_i)$ eines konventionellen Terms O_i, der ein Konstantensymbol O_i ist, mit dem formalen Objekt ob_u übereinstimmt, das aus der Anwendung $o_i()$ der Operation o_i auf das leere Argument hervorgeht. Die Operation o_i ist hierbei die extensionale Interpretation $I_{OPS}(O_i)$ von O_i in der SIG_{KS}-Struktur $A_{SIG_{KS}}$.

1) Die Interpretationsfunktion I_{RS} hat für die Auswertung von konventionellen Termen keine Bedeutung, da bei der Konstruktion von Termen keine Relationssymbole verwendet werden.

Bei der dritten Regel zur Auswertung konventioneller Terme handelt es sich um ein rekursives Schema. Die Auswertung eines zusammengesetzten konventionellen Terms $t \in ZT_{SIG_{KS}}$ setzt die Auswertung seiner (Term-)Bestandteile voraus. Ist ein Term im Argument eines zusammengesetzten Terms selbst ein zusammengesetzter Term, erfolgt ein erneuter Aufruf der Termauswertungsfunktion IT_{KS} mit dem letztgenannten Term im Argument. Dieses Rekursionsprinzip wird solange fortgesetzt, bis die „Abbruchbedingung" für die Rekursion erfüllt ist. Die Abbruchbedingung besteht darin, dass sich (mindestens) eine der beiden ersten Regeln zur Auswertung konventioneller Terme anwenden lässt. Die Abbruchbedingung für die Rekursion ist genau dann erfüllt, wenn der auszuwertende Term ein atomarer konventioneller Term ist. Die Menge $AT_{SIG_{KS}}$ aller atomaren konventionellen Terme entspricht der Menge $IND_{SIG_{KS}}$ aller Individuensymbole aus einer konventionellen Signatur SIG_{KS}. $IND_{SIG_{KS}}$ setzt sich wiederum aus der Variablenmenge VAR und der Menge $KON_{SIG_{KS}}$ aller Konstantensymbole zusammen. Wenn es sich bei dem auszuwertenden atomaren Term um eine Variable handelt, dann ist die Abbruchbedingung für die Rekursion entsprechend Regel (1.) erfüllt. In diesem Fall entspricht die Auswertung des Terms der Belegung der Variablen, die er darstellt. Wenn es sich bei dem auszuwertenden Term hingegen um ein Konstantensymbol handelt, ist die Abbruchbedingung entsprechend Regel (2.) erfüllt. Die Auswertung des atomaren Terms O_i ist dann die Anwendung $o_i()$ der Operation o_i auf das leere Argument, woraus das Individuum $ob_u \in OB$ hervorgeht.

2.2.1.1.3.2.2 Auswertung von konventionellen Formeln

Für die Auswertung von Formeln über einer konventionellen Signatur SIG_{KS} wird ihre *Bestätigung* in einer SIG_{KS}-Struktur $A_{SIG_{KS}}$ überprüft.[1] Dabei ist die Bestätigung einer konventionellen Formel immer in Bezug auf eine Variablenbelegung bel_{KS} definiert. Eine konventionelle Formel $F \in FORM_{SIG_{KS}}$ wird genau dann von einer Variablenbelegung bel_{KS} in einer SIG_{KS}-Struktur $A_{SIG_{KS}}$ bestätigt, wenn die Beziehung

$$(A_{SIG_{KS}}, bel_{KS}) \models F$$

gilt. In dem Ausdruck werden ein Tupel $(A_{SIG_{KS}}, bel_{KS})$ – bestehend aus einer SIG_{KS}-Struktur $A_{SIG_{KS}}$ und einer Variablenbelegung bel_{KS} – und eine konventionellen Formel $F \in FORM_{SIG_{KS}}$ in die Modellrelation $\models$ zueinander gesetzt. Dabei ist die Modellrelation $\models$ wie folgt rekursiv – über den Aufbau konventioneller Formeln – definiert:[2]

(1.) $(A_{SIG_{KS}}, bel_{KS}) \models w$ trifft für jedes beliebige Tupel $(A_{SIG_{KS}}, bel_{KS})$ zu,

(2.) $(A_{SIG_{KS}}, bel_{KS}) \models f$ trifft für kein Tupel $(A_{SIG_{KS}}, bel_{KS})$ zu,

(3.) $(A_{SIG_{KS}}, bel_{KS}) \models R_j(t_1,...,t_n) \quad \Leftrightarrow \quad (IT_{KS}(t_1),...,IT_{KS}(t_n)) \in r_j$,

1) Vgl. EHRIG ET AL. (1999), S. 320.

2) Vgl. GALLIER (1986), S. 169.

(4.) $(A_{SIG_{KS}},bel_{KS}) \models \neg F \quad \Leftrightarrow \quad (A_{SIG_{KS}},bel_{KS}) \not\models F$[1],

(5.) $(A_{SIG_{KS}},bel_{KS}) \models F_1 \wedge F_2 \quad \Leftrightarrow \quad (A_{SIG_{KS}},bel_{KS}) \models F_1$
und $(A_{SIG_{KS}},bel_{KS}) \models F_2$,

(6.) $(A_{SIG_{KS}},bel_{KS}) \models F_1 \vee F_2 \quad \Leftrightarrow \quad (A_{SIG_{KS}},bel_{KS}) \models F_1$
oder $(A_{SIG_{KS}},bel_{KS}) \models F_2$,

(7.) $(A_{SIG_{KS}},bel_{KS}) \models F_1 \underline{\vee} F_2 \quad \Leftrightarrow$ entweder $(A_{SIG_{KS}},bel_{KS}) \models F_1$
oder $(A_{SIG_{KS}},bel_{KS}) \models F_2$,

(8.) $(A_{SIG_{KS}},bel_{KS}) \models F_1 \rightarrow F_2 \quad \Leftrightarrow \quad (A_{SIG_{KS}},bel_{KS}) \not\models F_1)$
oder $(A_{SIG_{KS}},bel_{KS}) \models F_2$,

(9.) $(A_{SIG_{KS}},bel_{KS}) \models F_1 \leftrightarrow F_2 \quad \Leftrightarrow \quad (A_{SIG_{KS}},bel_{KS}) \models F_1 \rightarrow F_2$
und $(A_{SIG_{KS}},bel_{KS}) \models F_2 \rightarrow F_1$,

(10.) $(A_{SIG_{KS}},bel_{KS}) \models \forall x\colon F \quad \Leftrightarrow \quad (A_{SIG_{KS}},bel_{KS}[x/ob_u]) \models F$
für alle $ob_u \in OB$,

(11.) $(A_{SIG_{KS}},bel_{KS}) \models \exists x\colon F \quad \Leftrightarrow \quad (A_{SIG_{KS}},bel_{KS}[x/ob_u]) \models F$
für mindestens ein $ob_u \in OB$ und

(12.) $(A_{SIG_{KS}},bel_{KS}) \models \underline{\exists} x\colon F \quad \Leftrightarrow \quad (A_{SIG_{KS}},bel_{KS}[x/ob_u]) \models F$
für genau ein $ob_u \in OB$.

Wenn eine konventionelle Formel $F \in FORM_{SIG_{KS}}$ durch *jede* zulässige Variablenbelegung bel_{KS} in einer SIG_{KS}-Struktur $A_{SIG_{KS}}$ bestätigt wird, wird F als *gültig* in der SIG_{KS}-Struktur $A_{SIG_{KS}}$ bezeichnet. Die SIG_{KS}-Struktur $A_{SIG_{KS}}$ wird in diesem Fall als *Modell* der Formel F bezeichnet. Wenn eine Formel F in einer SIG_{KS}-Struktur $A_{SIG_{KS}}$ gültig ist, wird das fortan in der Form

$$A_{SIG_{KS}} \models F$$

angegeben.

Für die Ausweitung der Modellrelation $\models$ auf *Mengen* konventioneller Formeln wird eine implizite konjunktive Verknüpfung der Elemente der Formelmenge angenommen. Eine Formelmenge $FM=\{F_1,...,F_x\}$ kann demnach als die konjunktive Verknüpfung $FM'=F_1 \wedge ... \wedge F_X$ angenommen werden. So wird eine Formelmenge $FM \subseteq FORM_{SIG_{KS}}$ mit $FM=\{F_1,...,F_x\}$ genau dann durch eine Variablenbelegung bel_{KS} in einer SIG_{KS}-Struktur $A_{SIG_{KS}}$ bestätigt, wenn die Formel $F_1 \wedge ... \wedge F_x$ durch bel_{KS} in $A_{SIG_{KS}}$ bestätigt wird. Demnach gilt

$$(A_{SIG_{KS}},bel_{KS}) \models FM$$
mit $FM \subseteq FORM_{SIG_{KS}}$ und $FM=\{F_1,...,F_x\}$ genau dann, wenn gilt:
$$(A_{SIG_{KS}},bel_{KS}) \models F_1 \wedge ... \wedge F_x.$$

1) Der Ausdruck $(A_{SIG_{KS}},bel_{KS}) \not\models F$ stellt den zu der Modellrelation $\models$ komplementären Fall dar, in dem die konventionelle Formel F *nicht* von der Variablenbelegungsfunktion bel_{KS} in der SIG_{KS}-Struktur $A_{SIG_{KS}}$ bestätigt wird.

Die Formelmenge FM ist genau dann in einer SIG_{KS}-Struktur $A_{SIG_{KS}}$ gültig, wenn die konjunktive Verknüpfung $F_1 \wedge ... \wedge F_x$ aller Formeln $F_1,...,F_x \in FM$ in $A_{SIG_{KS}}$ gültig ist. Auch hierfür wird die Schreibweise wie für einzelne konventionelle Formeln übernommen:

$$A_{SIG_{KS}} \vDash FM$$
$$\text{mit } FM \subseteq FORM_{SIG_{KS}} \text{ und } FM=\{F_1,...,F_x\} \text{ genau dann, wenn gilt:}$$
$$A_{SIG_{KS}} \vDash F_1 \wedge ... \wedge F_x.$$

Mit der Modellrelation $\vDash$ wird eine Beziehung zwischen einer SIG_{KS}-Struktur $A_{SIG_{KS}}$ und einer Variablenbelegung bel_{KS} einerseits und konventionellen Formeln oder Formelmengen andererseits ausgedrückt. Die Modellrelation $\vDash$ hat insofern einen *semantischen* Charakter, als dass ihre Elemente von den extensionalen Interpretationsfunktionen aus I_{KS} abhängen. Die extensionalen Interpretationsfunktionen bilden den Kern der formalen Semantik, die der konventionellen Prädikatenlogik zugrunde liegt. Darauf basierend wird die Auswertung konventioneller Terme durchgeführt, die für die Auswertung konventioneller Formeln nötig ist.

Die Menge aller Modelle zu einer Formel $F \in FORM_{SIG_{KS}}$ wird durch die Funktion

$$MOD_{KS}: FORM_{SIG_{KS}} \rightarrow pot(A(SIG_{KS}))$$
$$\text{mit } MOD_{KS}(F) = \{A_{SIG_{KS}} \mid (A_{SIG_{KS}} \vDash F)\}$$

angegeben.[1)] Zu einer Formel $F \in FORM_{SIG_{KS}}$ ist $MOD_{KS}(F)$ die Menge aller SIG_{KS}-Strukturen, in denen die Formel F gültig ist. Die gleiche Schreibweise wird im Folgenden auch für Formelmengen beibehalten. Demnach gilt für eine Formelmenge $FM \subseteq FORM_{SIG_{KS}}$ mit $FM=\{F_1,...,F_x\}$:

$$MOD_{KS}(FM) = \{A_{SIG_{KS}} \mid (A_{SIG_{SS}} \vDash (F_1 \wedge ... \wedge F_x))\ \}$$
$$\text{oder } MOD_{KS}(FM)=MOD_{KS}(F_1 \wedge ... \wedge F_x).$$

Die Menge $MOD_{KS}(FM)$ aller Modelle einer Formelmenge FM umfasst alle SIG_{KS}-Strukturen, in denen die Formel $F_1 \wedge ... \wedge F_x$, die aus der konjunktiven Verknüpfung aller Formeln aus FM hervorgeht, gültig ist.

Bei einer Vereinigung $FM_1 \cup FM_2$ von zwei Formelmengen $FM_1 \subseteq FORM_{SIG_{KS}}$ und $FM_2 \subseteq FORM_{SIG_{KS}}$ entspricht die Menge $MOD_{KS}(FM_1 \cup FM_2)$ der Schnittmenge $MOD_{KS}(FM_1) \cap MOD_{KS}(FM_2)$:[2)]

$$\forall FM_1,FM_2 \subseteq FORM_{SIG_{KS}}: MOD_{KS}(FM_1 \cup FM_2) = MOD_{KS}(FM_1) \cap MOD_{KS}(FM_2).$$

Die Gültigkeit der Formelmenge $FM_1 \cup FM_2$ ist deswegen auf die Schnittmenge der beiden Modellmengen beschränkt, weil nur solche SIG_{KS}-Strukturen als Modell in Frage kommen, in denen alle Formeln aus $FM_1 \cup FM_2$ gültig sind.

1) Vgl. EHRIG ET AL. (1999), S. 335.

2) Vgl. EHRIG ET AL. (1999), S. 334.

Wenn eine Formelmenge $FM_1 \subseteq FORM_{SIG_{KS}}$ eine Teilmenge $FM_1 \subseteq FM_2$ einer Formelmenge $FM_2 \subseteq FORM_{SIG_{KS}}$ ist, dann ist die Menge $MOD_{KS}(FM_2)$ aller Modelle der Formelmenge FM_2 eine Teilmenge der Menge $MOD_{KS}(FM_1)$ aller Modelle der Formelmenge FM_1:[1)]

$$\forall FM_1, FM_2 \subseteq FORM_{SIG_{KS}}: FM_1 \subseteq FM_2 \rightarrow MOD_{KS}(FM_2) \subseteq MOD_{KS}(FM_1).$$

Eine Formel $F \in FORM_{SIG_{KS}}$ ist genau dann *erfüllbar*, wenn es *mindestens eine* SIG_{KS}-Struktur $A_{SIG_{KS}} \in A(SIG_{KS})$ gibt, in der F von mindestens einer Variablenbelegung bel_{KS} bestätigt wird. Eine Formelmenge $FM \subseteq FORM_{SIG_{KS}}$ mit $FM=\{F_1,...F_x\}$ ist genau dann erfüllbar, wenn die konjunktive Verknüpfung $F_1 \wedge ... \wedge F_x$ aller Formeln aus FM erfüllbar ist.

Die Formel $F \in FORM_{SIG_{KS}}$ ist *allgemeingültig* oder *tautologisch*, wenn sie von *jeder* Variablenbelegung bel_{KS} in *jeder* SIG_{KS}-Struktur $A_{SIG_{KS}} \in A(SIG_{KS})$ bestätigt wird.[2)] Für eine allgemeingültige Formel $F \in FORM_{SIG_{KS}}$ über einer konventionellen Signatur SIG_{KS} gilt, dass die Menge $A(SIG_{KS})$ aller SIG_{KS}-Strukturen sich mit der Menge $MOD_{KS}(F)$ aller Modelle von F deckt. Demnach gilt für eine allgemeingültige Formel $F \in FORM_{SIG_{KS}}$:

$$A(SIG_{KS}) = MOD_{KS}(F).$$

Eine konventionelle Formel $F \in FORM_{SIG_{KS}}$ ist *widersprüchlich* oder *kontradiktorisch*, wenn keine SIG_{KS}-Struktur $A_{SIG_{KS}}$ und keine Variablenbelegung bel_{KS} existieren, so dass F durch bel_{KS} in $A_{SIG_{KS}}$ bestätigt wird. Für jede widersprüchliche Formel über einer konventionellen Signatur SIG_{KS} gilt, dass es keine SIG_{KS}-Struktur $A_{SIG_{KS}}$ gibt, die ein Modell der widersprüchlichen Formel wäre.[3)] Eine Formelmenge $FM \subseteq FORM_{SIG_{KS}}$ ist genau dann widersprüchlich, wenn die konjunktive Verknüpfung $F_1 \wedge ... \wedge F_x$ aller Formeln aus FM eine widersprüchliche Formel ist.

---

1) Vgl. EHRIG ET AL. (1999), S. 334.

2) Beispielsweise ist die Formel

$$R_j(x_q) \vee \neg R_j(x_q)$$

unabhängig von der Belegung $bel_{KS}(x_q)$ der Variablen x_q und der entsprechenden Termauswertung $IT_{KS}=bel_{KS}(x_q)$ in jeder SIG_{KS}-Struktur $A_{SIG_{KS}} \in A(SIG_{KS})$ gültig. Mit der aufgeführten Beispieltautologie wird auch das Prinzip des *tertium non datur* verdeutlicht. Dabei handelt es sich um das Prinzip, dass entweder eine Formel oder die Negation der gleichen Formel gültig sein muss. Eine dritte Gültigkeitsvariante wird ausgeschlossen. Daher ist eine Formel eine Tautologie, wenn sie aus zwei adjunktiv verknüpften Teilformeln besteht, wobei ein Adjunktionsglied die Negation des anderen Adjunktionsglieds ist.

3) Die Formel

$$R_j(x_q) \wedge \neg R_j(x_q)$$

ist z.B. widersprüchlich, weil sie unabhängig von der Variablenbelegung $bel_{KS}(x_q)$ der Variable x_q und der entsprechenden Termauswertung $IT_{KS}(x_q)=bel_{KS}(x_q)$ in keiner SIG_{KS}-Struktur $A_{SIG_{KS}}$ gültig sein kann.

Mit der Modellrelation $\vDash$ werden Aussagen über die grundsätzliche Erfüllbarkeit einer Formel bzw. Formelmenge gemacht. Die Modellrelation $\vDash$ nimmt Bezug auf die Semantik einer Formel(-menge), um ihre Erfüllbarkeit in einer SIG_{KS}-Struktur bei Variablenbelegungen zu bestimmen. Um Aussagen darüber machen zu können, ob die Gültigkeit von Formeln in einer SIG_{KS}-Struktur aufgrund der Gültigkeit anderer Formeln gegeben sein muss, wird die *Folgerungsrelation* $\Vdash$ eingeführt. Während die Modellrelation $\vDash$ auf die Gültigkeit von Formeln in SIG_{KS}-Strukturen ausgerichtet ist, ist die Folgerungsrelation $\Vdash$ auf die *Folgerbarkeit* der Gültigkeit von Formeln aus der Gültigkeit von anderen Formeln ausgerichtet.

Genau dann, wenn die Menge $MOD_{KS}(F_1)$ aller Modelle einer Formel F_1 eine Teilmenge der Modelle $MOD_{KS}(F_2)$ einer Formel F_2 ist, folgt die Formel F_2 aus der Formel F_1. Für diesen Zusammenhang wird die Schreibweise $F_1 \Vdash F_2$ verwendet:

$$\forall F_1, F_2 \in FORM_{SIG_{KS}}: (MOD_{KS}(F_1) \subseteq MOD_{KS}(F_2)) \leftrightarrow (F_1 \Vdash F_2).$$

Die Folgerungsrelation $\Vdash$ wird – analog zu der Modellrelation $\vDash$ – auf Formelmengen ausgeweitet. Demnach folgt eine Formelmenge $F_2 \subseteq FORM_{SIG_{KS}}$ genau dann aus einer Formelmenge $F_1 \subseteq FORM_{SIG_{KS}}$, wenn die Menge $MOD_{KS}(F_1)$ aller Modelle von F_1 eine Teilmenge der Modelle $MOD_{KS}(F_2)$ von F_2 ist:

$$\forall FM_1, FM_2 \subseteq FORM_{SIG_{KS}}: (MOD_{KS}(FM_1) \subseteq MOD_{KS}(FM_2)) \leftrightarrow (FM_1 \Vdash FM_2).$$

Die Folgerungsrelation $\Vdash$ kann als eine *Quasiordnung* charakterisiert werden.[1)] Die Folgerungsrelation $\Vdash$ ist nämlich sowohl reflexiv als auch transitiv. Erstens gilt, dass jede Formel $F \in FORM_{SIG_{KS}}$ stets mit sich selbst in einer Folgerungsbeziehung $\Vdash$ steht (Reflexivität):

$$\forall F \in FORM_{SIG_{KS}}: (F \Vdash F).$$

Zweitens gilt, dass, wenn eine Formel $F_1 \in FORM_{SIG_{KS}}$ in Folgerungsbeziehung mit einer Formel $F_2 \in FORM_{SIG_{KS}}$ steht, die wiederum mit einer Formel $F_3 \in FORM_{SIG_{KS}}$ in einer Folgerungsbeziehung steht, die Formel F_1 auch in einer Folgerungsbeziehung mit der Formel F_3 steht (Transitivität):

$$\forall F_1,F_2,F_3:FORM_{SIG_{KS}}: ((F_1 \Vdash F_2) \land (F_2 \Vdash F_3)) \rightarrow (F_1 \Vdash F_3).$$

Die Transitivität der Folgerungsrelation ist mit der Transitivität der Teilmengenrelation ($\subseteq$) begründet, die die Mengen $MOD_{KS}(F_1)$, $MOD_{KS}(F_2)$ und $MOD_{KS}(F_3)$ verbindet.

Ein weiterer wesentlicher Aspekt der Folgerungsrelation $\Vdash$ ist ihre *Monotonie*. Sie äußert sich in zwei Eigenarten von $\Vdash$. Zum einen folgt eine Formelmenge FM_2 auch aus jeder Obermenge FM_3 einer Formelmenge FM_1, wenn FM_2 aus FM_1 folgt:

$$\forall FM_1,FM_2,FM_3 \subseteq FORM_{SIG_{KS}}: ((FM_1 \Vdash FM_2) \land (FM_1 \subseteq FM_3)) \rightarrow (FM_3 \Vdash FM_2).$$

Zum anderen folgt jede Teilmenge einer Formelmenge FM_2 aus einer Formelmenge FM_1, wenn FM_2 aus FM_1 folgt:

1) Vgl. EHRIG ET AL. (1999), S. 342.

$$\forall FM_1,FM_2,FM_3 \subseteq FORM_{SIG_{KS}}: ((FM_1 \Vdash FM_2) \wedge (FM_3 \subseteq FM_2)) \rightarrow (FM_1 \Vdash FM_3).$$

Für die Bestimmung der Folgerbarkeit von Formeln untereinander kann auf eine Vielzahl *metasprachlicher Inferenzregeln*[1] zurückgegriffen werden. Unter Bezugnahme auf die Form objektsprachlicher Formeln erlauben metasprachliche Inferenzregeln die Ableitung objektsprachlicher Formeln aus objektsprachlichen Formeln. Eine Gesamtheit metasprachlicher Inferenzregeln wird als *Inferenzkalkül* bezeichnet. Als Anforderungen an Inferenzkalküle gelten die *Vollständigkeit* und *Korrektheit*.[2] Ermöglicht das Inferenzkalkül die Ableitung *aller* folgerbaren Ableitungen, so wird es als *vollständig* bezeichnet. Ist das Inferenzkalkül auf solche Ableitungen beschränkt, die folgerbare Formeln hervorbringen, so wird es als *korrekt* bezeichnet.

Die bekanntesten metasprachlichen Inferenzregeln sind der *modus ponens* und der *modus tollens*.[3] Der modus ponens erlaubt die Ableitung der Konklusions-Formel F_2, wenn die Antezedenz-Formeln F_1 und $F_1 \rightarrow F_2$ vorliegen. Der modus tollens erlaubt hingegen die Ableitung der Konklusions-Formel $\neg F_1$, wenn die Antezedenz-Formeln $\neg F_2$ und $F_1 \rightarrow F_2$ vorliegen. Beide metasprachlichen Inferenzregeln nehmen in ihren Antezedenz-Komponenten u.a. eine Subjugats-Formel auf.

Es ist nicht ausgeschlossen, dass zwei unterschiedliche Formeln $F_1,F_2 \in FORM_{SIG_{KS}}$ wechselseitig in der Folgerungsrelation $\Vdash$ zueinander stehen. Wenn sowohl $(F_1 \Vdash F_2)$ als auch $(F_2 \Vdash F_1)$ gelten, sind die Mengen $MOD_{KS}(F_1)$ und $MOD_{KS}(F_1)$ gleich. In diesem Fall werden die Formeln F_1 und F_2 als *logisch äquivalent* bezeichnet, und ihr Verhältnis wird in der Form $F_1 \equiv F_2$ angegeben:

$$\forall F_1,F_2 \in FORM_{SIG_{KS}}: ((F_1 \Vdash F_2) \wedge (F_2 \Vdash F_2)) \leftrightarrow (F_1 \equiv F_2)$$

Die logische Äquivalenz ist ebenso übertragbar auf Formelmengen. Eine Formelmenge $FM_1 \subseteq FORM_{SIG_{KS}}$ ist logisch äquivalent mit einer Formelmenge $FM_2 \subseteq FORM_{SIG_{KS}}$, wenn die Menge $MOD_{KS}(FM_1)$ aller Modelle der Formelmenge FM_1 gleich ist mit der Menge $MOD_{KS}(FM_2)$ aller Modelle der Formelmenge FM_2.

In der Tabelle 3 sind die Grundformen logischer Äquivalenzen für prädikatenlogische Formeln wiedergegeben.[4]

1) Die Bezeichnung „Inferenzregel" wird im weiteren Verlauf dieser Arbeit auf objektsprachliche Konstrukte bezogen. Objektsprachliche Inferenzregeln werden in Abschnitt 3.1.4.1 näher behandelt

2) Vgl. EHRIG ET AL. (1999), S. 269; HERMES (1991), S. 84 f.; SCHÖNING (1992), S. 39.

3) Vgl. KREOWSKI (1991), S. 114 ff.

4) Vgl. EHRIG ET AL. (1999), S. 239.

Implikation	$(F_1 \rightarrow F_2)$	$\equiv$	$\neg F_1 \lor F_2$
DeMorgan-Gesetze	$\neg(F_1 \lor F_2)$	$\equiv$	$\neg F_1 \land \neg F_2$
	$\neg(F_1 \land F_2)$	$\equiv$	$\neg F_1 \lor \neg F_2$
Doppelte Negation	$\neg\neg F_1$	$\equiv$	F_1
Idempotenzgesetze	$F_1 \lor F_1$	$\equiv$	F_1
	$F_1 \land F_1$	$\equiv$	F_1
Absorptionsgesetze	$F_1 \land (F_1 \lor F_2)$	$\equiv$	F_1
	$F_1 \lor (F_1 \land F_2)$	$\equiv$	F_1
Kommutativgesetze	$F_1 \land F_2$	$\equiv$	$F_2 \land F_1$
	$F_1 \lor F_2$	$\equiv$	$F_2 \lor F_1$
Assoziativgesetze	$F_1 \land (F_2 \land F_3)$	$\equiv$	$(F_1 \land F_2) \land F_3$
	$F_1 \lor (F_2 \lor F_3)$	$\equiv$	$(F_1 \lor F_2) \lor F_3$
Distributivgesetze	$F_1 \land (F_2 \lor F_3)$	$\equiv$	$(F_1 \land F_2) \lor (F_1 \land F_3)$
	$F_1 \lor (F_2 \land F_3)$	$\equiv$	$(F_1 \lor F_2) \land (F_1 \lor F_3)$

Tabelle 3: Logische Äquivalenzen

Bei den oben aufgeführten logischen Äquivalenzen handelt es sich um solche Äquivalenzen, die aussagenlogischen Charakter haben. Für die oben aufgeführten Formeln $F_1,F_2,F_3 \in FORM_{SIG_{KS}}$ ließen sich nämlich auch aussagenlogische Formeln einsetzen, wodurch die Äquivalenzen beibehalten würden.

Für die Prädikatenlogik sind darüber hinaus noch weitere Äquivalenzen bekannt. Beispielsweise gilt für die beiden Formeln $F_1,F_2 \in FORM_{SIG_{KS}}$ mit $F_1=\forall x{:}F(x)$ und $F_2=\neg\exists x{:}\neg F(x)$ auch $F_1 \equiv F_2$.

2.2.1.2 Sortierte Prädikatenlogik

2.2.1.2.1 Überblick über die sortierte Prädikatenlogik

In einer konventionellen Signatur SIG_{KS} sind als metasprachliche Ausdrucksmittel nur die Typisierungsfunktionen $typ_{OPS_{KS}}$ und $typ_{RS_{KS}}$ zugelassen. Während mit der Typisierungsfunktion $typ_{OPS_{KS}}$ die Stelligkeit von Operationssymbolen angegeben wird, wird die Typisierungsfunktion $typ_{RS_{KS}}$ dazu verwendet, die Stelligkeit von Relationssymbolen zu spezifizieren. Beide Typisierungsfunktionen bilden jeweils Operations- bzw. Relationssymbole auf eine natürliche Zahl ab, die der *Anzahl* der konventionellen Terme entspricht, die das entsprechende Symbol in seinem Argument aufnehmen kann, um einen zulässigen Ausdruck formieren zu können. Dadurch werden als zulässige Ausdrücke solche Konstruktionen ausgeschlossen, die nicht mit der Typisierung der verwendeten Operations- und Relationssymbole vereinbar sind. Beispielsweise kann für das Relationssymbol *kooperiert_mit* aus einer konventionellen Signatur SIG_{KS} der Typ $typ_{RS_{KS}}$(kooperiert_mit)=2 angegeben werden. Dadurch wird festgelegt, dass das Relationssymbol *kooperiert_mit* genau zwei Terme in seinem Argument aufnehmen muss, um eine atomare konventionelle Formel darzustellen. Sämtliche Konstruktionen, bei denen das Relationssymbol *kooperiert_mit* nicht genau zwei Terme in seinem Argument aufweist, werden nicht als konventionelle Formeln zugelassen.

Bei der Konstruktion von konventionellen Formeln ist es vollkommen irrelevant, welche *Arten von Termen* in den Argumenten von Relationssymbolen verwendet werden, um atomare konventionelle Formeln zu konstruieren. Für die Menge $TERM_{SIG_{KS}}$ aller konventionellen Terme ist nämlich keine Unterteilung hinsichtlich der Art konventioneller Terme vorgesehen. Dieser undifferenzierten Termmenge entspricht auf der Seite einer SIG_{KS}-Struktur $A_{SIG_{KS}}$, mit der eine konventionelle Signatur SIG_{KS} extensional interpretiert wird, eine Objektmenge OB, deren Elemente auch nicht hinsichtlich ihrer Art differenziert werden. Somit ist z.B. der Ausdruck

kooperiert_mit(U_1,Drei)

in der konventionellen Prädikatenlogik als Formel zugelassen, obwohl er in jeder SIG_{KS}-Struktur ungültig sein muss. Die Ungültigkeit der Formel ist allerdings nicht durch ihre Form bedingt, sondern durch ihren *Sinn*. Es macht nämlich keinen Sinn, dass ein *Unternehmen* U_1 mit der *Zahl* Drei *kooperiert*. Die Aussage würde dann Sinn machen, wenn mit der Bezeichnung „Drei" ein Unternehmen gemeint wäre. In der konventionellen Prädikatenlogik ist kein Verfahren vorgesehen, mit dem solche Formeln bereits durch die Grammatiken zur Konstruktion von Aussagen ausgeschlossen werden könnten. Grammatiken dieser Art werden erst im Rahmen der sortierten Prädikatenlogik bereitgestellt.

Mit dem Übergang zur *sortierten Prädikatenlogik*[1)] sind vielfache Vorteile bei der prädikatenlogischen Wissensrepräsentation gegenüber der konventionellen Prädikatenlogik verbunden. Durch die sortenspezifische Zuordnung von Termen können Aussagen, bei deren Konstruktion auf *sortenspezifische Terme* zurückgegriffen wird, aufgrund der Unmöglichkeit, ihnen einen *Sinn* zuzuweisen, ausgeschlossen werden. Dadurch werden bereits in der Grammatik zur Konstruktion von Aussagen Konstrukte ausgeschlossen, die sinnlos sind. Darüber hinaus erweist sich die sortierte Prädikatenlogik gegenüber der konventionellen Prädikatenlogik insofern als „effizienter", als dass der Suchraum für Variablen, über die sich Quantoren erstrecken, eingeschränkt wird.

Die Kollision von *Bedeutung*[2)] und *Sinn*[3)] einer prädikatenlogischen Aussage kann durch die sortenspezifische Zuordnung von Objekten zumindest ansatzweise[4)] angegangen werden. Bei allen prädikatenlogischen Aussagen stellt sich nämlich vor der Frage ihrer Bedeutung stets die Frage nach ihrer Sinnhaftigkeit. Bei der sortierten Prädikatenlogik kann beispielsweise im Vorfeld bestimmt werden, dass Kooperationsverhältnisse lediglich zwischen Unternehmen zulässig sind. Dadurch wird die o.a. Aussage im Vorfeld wegen ihrer Sinnlosigkeit ausgeschlossen. Die Frage nach der Bedeutung der Aussage wird dann zweitrangig, weil die Aussage sinnlos ist. Aufgrund ihres fehlenden Sinns gilt die Aussage als syntaktisch unzulässig und kann entsprechend gar nicht konstruiert werden. Der syntaktische Ausschluss sinnloser Formeln ist somit semantisch motiviert. Bereits bei der Konstruktion von Aussagen werden solche Formeln, die gar keine Bedeutung haben können, weil sie keinen Sinn haben, ausgeschlossen.[5)] Denn einer sinnlosen Formel kann auch keine Bedeutung zugewiesen werden.

Die sortierte Prädikatenlogik beinhaltet jedoch keine Ausweitung der Ausdrucksmöglichkeiten gegenüber der konventionellen Prädikatenlogik. Wie später aufgezeigt wird, können nämlich alle Ausdrücke, die in einer sortierten Prädikatenlogik formuliert wer-

1) Für einen Überblick zur sortierten Prädikatenlogik vgl. EHRIG ET AL. (1999), S. 307 ff.; GUESSERIAN (1993), S. 124 ff.; KREOWSKI (1991), S. 33 ff.; LOECKX ET AL. (1996), S. 78 ff.; MANZANO (1993), S. 3 ff.; OBERSCHELP (1962), S. 297 ff. Für eine Umsetzung eines konkreten Spezifikationsansatzes für das Produktionsmanagement mittels sortierter Prädikatenlogik vgl. PATIG (2001), S. 53 ff.

2) Der Begriff *Bedeutung* wird im Folgenden als Unterbegriff zum Begriff *Semantik* verwendet. Mit *Bedeutung* wird der *extensionale* Aspekt von Semantik angesprochen. Daher könnten im Folgenden die Bezeichnungen *Bedeutung* und *Extension* auch synonym verwendet werden. Der Verfasser bevorzugt jedoch die Bezeichnung Extension, da dies der etablierten Terminologie entspricht.

3) Der *intensionale* Aspekt einer Semantik wird durch den zweiten Unterbegriff *Sinn* erfasst. Daher können im Folgenden die Begriffe *Intension* und *Sinn* synonym verwendet werden.

4) Im weiteren Verlauf wird aufgezeigt, welche Probleme hinsichtlich der Erfassung der intensionalen Semantik auch durch die sortenspezifische Zuordnung von Objektmengen weiterhin ungelöst bleiben.

5) FREGE bezeichnet den Sinn von Aussagen auch als „Gedanke"; vgl. FREGE (1966), S. 33. An gleicher Stelle formuliert FREGE: „... nenne ich Gedanken etwas, bei dem überhaupt Wahrheit in Frage kommen kann".

den, auch in der konventionellen Prädikatenlogik formuliert werden.[1)] Die Bereicherung gegenüber der konventionellen Prädikatenlogik liegt in erster Linie im „Formulierungskomfort“[2)] bei der Konstruktion sortierter prädikatenlogischer Aussagen.

In der sortierten Prädikatenlogik wird die Menge aller Terme in *sortenspezifische* Termmengen unterteilt[3)]. Die Unterteilung betrifft sowohl Terme, die aus Individuensymbolen hervorgehen, als auch Terme, die aus der Anwendung von Operationssymbolen auf andere Terme hervorgehen. Dadurch können sowohl für einfache als auch für zusammengesetzte Terme Sortenzugehörigkeiten angegeben werden. Beachtenswert ist dabei, dass bei der Konstruktion zusammengesetzter Terme wiederum die Sortenzugehörigkeiten der herangezogenen Terme berücksichtigt werden. Die Interpretation der Terme aus sortenspezifischen Termmengen erfolgt in diesem Fall mittels formaler Objekte aus wiederum sortenspezifischen Objektmengen. Der sortenspezifisch differenzierten Menge aller Terme über einer sortierten Signatur steht somit eine sortenspezifisch differenzierte Menge formaler Objekte gegenüber.

Die – im Vergleich zur konventionellen Prädikatenlogik – differenzierte Handhabung objektsprachlicher Ausdrucksmittel im Rahmen der sortierten Prädikatenlogik trifft auch auf Variablen zu. Auch für Variablen werden ihre Zugehörigkeiten zu Sorten spezifiziert. In ihrer Verwendung als Terme werden die zugewiesenen Sorten für Variablen beibehalten.

Darüber hinaus ist ein weiterer Aspekt für sortenspezifische Variablen von Bedeutung. Im Rahmen der konventionellen Prädikatenlogik werden alle Individuen aus dem gesamten prädikatenlogischen Universum OB zur Auswertung quantifizierter Formeln überprüft. Wird beispielsweise in einer konventionellen Formel $F \in FORM_{SIG_{KS}}$ der Allquantor $\forall$ verwendet, so muss die durch $\forall$ gebundene Variable mit *allen* formalen Objekten aus dem prädikatenlogischen Universum OB belegt werden, um die Gültigkeit von F in einer SIG_{KS}-Struktur $A_{SIG_{KS}}$ zu überprüfen. Entsprechend kann es zur „explosionsartigen“ Vergrößerung des Suchraums kommen, wenn die Gültigkeit quantifizierter Formeln überprüft werden soll. In diesem Punkt erweist sich die sortierte Prädikatenlogik als wesentlich effizienter.[4)] Der Suchraum für Variablen wird im Rahmen der sor-

1) Auf diesen Aspekt wird in Kürze näher eingegangen werden.

2) ZELEWSKI (1995), Bd. 4, S. 227.

3) Die konventionelle Prädikatenlogik kann auch als ein Sonderfall der sortierten Prädikatenlogik charakterisiert werden; vgl. MANZANO (1993), S. 4; EHRIG ET AL. (1999), S. 311. Die Erweiterung durch die sortierte Prädikatenlogik liegt demnach darin, dass die konventionelle Prädikatenlogik nur *eine* Sorte betrachtet. Die Sortenmenge S der konventionellen Prädikatenlogik hat daher stets die Mächtigkeit $|S|=1$. Daher wird die Menge aller Sorten für konventionelle Signaturen nicht angegeben. In der sortierten Prädikatenlogik ist hingegen grundsätzlich eine Mächtigkeit der Sortenmenge $|S|\geq 1$ zugelassen. Darüber hinaus kann jede Sorte aus einer sortierten Signatur auch als einstelliges Relationssymbol mit entsprechender Interpretation durch eine Relation betrachtet werden; vgl. BEIERLE ET AL. (1993), S. 192. Auf diesen Aspekt wird im nächsten Abschnitt näher eingegangen.

4) Vgl. ZELEWSKI (1995), Bd. 4, S. 224.

tierten Prädikatenlogik auf jene Objektmenge eingeschränkt, die der gleichen Sorte zugeordnet ist, wie die durch den Quantor gebundene Variable. Dadurch können Inferenzkonzepte der sortierten Prädikatenlogik im Allgemeinen wesentlich schneller durchgeführt werden als in der konventionellen Prädikatenlogik.

2.2.1.2.2 Syntaktische Aspekte der sortierten Prädikatenlogik

2.2.1.2.2.1 Sortierte Signaturen

Die syntaktischen Aspekte der sortierten Prädikatenlogik erstrecken sich zum einen auf ein formalsprachliches Alphabet $ALPH_{SS}$. In dem formalsprachlichen Alphabet $ALPH_{SS}$ sind alle Symbole enthalten, die zur Konstruktion von Formeln über einer sortierten Signatur SIG_{SS} zugelassen sind. Dabei geht das Alphabet $ALPH_{SS}$ aus dem Alphabet $ALPH_{KS}$ dadurch hervor, dass Letztgenanntes um eine Menge S von *Sorten* erweitert wird:[1)]

$$ALPH_{SS} = ALPH_{KS} \cup S$$
$$\text{mit } S = \{s_1,...,s_n\} \text{ mit } n=1,...,N \text{ und } N \in \mathbb{N}_+.$$

Somit unterscheidet sich das formalsprachliche Alphabet $ALPH_{SS}$ für die sortierte Prädikatenlogik von dem formalsprachlichen Alphabet $ALPH_{KS}$ dadurch, dass die Menge S aller Sorten hinzugekommen ist. Die Funktionalität von Sorten wird im Kontext sortierter Signaturen in den nächsten Abschnitten vorgestellt. Im Übrigen stimmen die formalsprachlichen Alphabete $ALPH_{KS}$ und $ALPH_{SS}$ der konventionellen bzw. sortierten Prädikatenlogik überein. Beide Alphabete umfassen die gleiche Menge logischer Symbole. Ebenso sind für beide Alphabete die gleichen Operations- und Relationssymbole, Variablen und Hilfssymbole vorgesehen.

Als Ausdrücke über dem formalsprachlichen Alphabet $ALPH_{SS}$ kommen Elemente der Menge $ALPH_{SS}*$ aller Wörter über $ALPH_{SS}$ in Betracht. Um durch eine Grammatik bestimmen zu können, welche Wörter über $ALPH_{SS}$ auch als Ausdrücke zugelassen sind, müssen die Operations- und Relationssymbole aus $ALPH_{SS}$ in einer sortierten Signatur *typisiert* werden.

Eine *sortierte Signatur* SIG_{SS} ist definiert als:

$$SIG_{SS}=(S,OPS,RS,VARF_{SIG_{SS}},typ_{OPS_{SS}},typ_{RS_{SS}}).$$

Die Komponenten einer sortierten Signatur SIG_{SS} sind:

(1.) Eine Menge $S = \{s_n\}$ von *Sorten* mit $n \in \{1,...,N\}$ und $N \in \mathbb{N}_+$,

(2.) eine Menge $OPS = \{O_i\}$ von *Operationssymbolen* mit $i \in \{1,...,I\}$ und $I \in \mathbb{N}$,

(3.) eine Menge $RS = \{R_j\}$ von *Relationssymbolen* mit $j \in \{1,...,J\}$ und $J \in \mathbb{N}$,

(4.) eine Familie $VARF_{SIG_{SS}} = (VAR_s)_{s \in S}$ sortenspezifischer *Variablenmengen*,

---

1) Die Entscheidung des Verfasser, die Sortenmenge S in das sortierte Alphabet $ALPH_{SS}$ aufzunehmen, kann nur gerechtfertigt werden, wenn $ALPH_{SS}$ nicht nur als die Gesamtheit aller objektsprachlichen Symbole aufgefasst wird, die auch in Ausdrücken vorkommen können. Denn Sorten tauchen weder in sortierten Termen noch in sortierten Formeln auf. Sie werden lediglich in der Grammatik zur Konstruktion von allen sortierten Ausdrücken berücksichtigt. Der vorliegenden Arbeit liegt jedoch das Verständnis zugrunde, Sorten ebenso als objektsprachliche Konstrukte zu betrachten, auch wenn sie nicht in objektsprachlichen Aussagen vorkommen.

(5.) eine Operationssymbol-Typisierungsfunktion $typ_{OPS_{SS}}$: OPS $\rightarrow$ $S^* \times S$ und

(6.) eine Relationssymbol-Typisierungsfunktion $typ_{RS_{SS}}$: RS $\rightarrow$ S^+.

Mit $S=\{s_1,...,s_N\}$ wird die Menge der *Sorten* bezeichnet, die im Rahmen einer formalen Semantik durch sortenspezifische Objektmengen extensional interpretiert werden. S entspricht selbst einem formalsprachlichen Alphabet, dessen Elemente $s_1,...,s_N$ als einstellige *Wörter* über S konzeptualisiert werden. Mit S* wird die Menge aller null-, ein- oder mehrstelligen Wörter bezeichnet, die über der Menge S konstruiert werden können. Jede Konkatenation $w \in S^*$ ist somit auch ein Wort über der Sortenmenge S. S* hat als Sonderfall auch das *leere Wort* λ als Element. Die Menge S^+ umfasst hingegen alle Wörter über der Sortenmenge S außer dem leeren Wort λ.

Wie bereits zuvor angesprochen, bietet die sortierte Prädikatenlogik im Vergleich zur konventionellen Prädikatenlogik keine Erweiterung der Ausdrucksmöglichkeiten. Die Menge S aller Sorten lässt sich nämlich auf die Menge RS der Relationssymbole zurückführen. Jede Sorte $s \in S$ kann demnach als ein einstelliges Relationssymbol $s \in RS$ aufgefasst werden.[1)] Im Unterschied zu den Relationssymbolen, die in der Menge RS aufgeführt werden, müsste allerdings für Relationssymbole, die aus Sorten hervorgehen, bestimmt werden, dass sie nicht beliebig typisiert[2)] werden können. Relationssymbole, die aus Sorten hervorgehen, dürfen nur als einstellige Relationssymbole typisiert werden. Mit der Rückführbarkeit von Sorten auf einstellige Relationssymbole ist auch die Rückführbarkeit der sortierten Prädikatenlogik auf die konventionelle Prädikatenlogik bewiesen.

Den Operationssymbolen $O_i \in OPS$ aus einer sortierten Signatur SIG_{SS} werden durch die Operationssymbol-Typisierungsfunktion[3)]:

$$typ_{OPS_{SS}}: OPS \rightarrow S^* \times S$$

eine *Stelligkeit* und eine *Sortenstruktur* zugewiesen. Für ein Operationssymbol $O_i \in OPS$ mit $typ_{OPS_{SS}}(O_i)=(s_1...s_n,s_{n+1})$ werden die Sorten $s_1...s_n$ als *Argumentsorten* und die Sorte s_{n+1} als *Zielsorte* bezeichnet. Hierbei ist n die Stelligkeit und $(s_1...s_n,s_{n+1})$ die Sortenstruktur des Operationssymbols O_i. Die Sortenkette $s_1...s_n$ kann auch als das Wort $w=s_1...s_n$ angesprochen werden. Für den formalen Zugriff auf Argument- und Zielsorten von Operationssymbolen werden die *Argumentfunktion*

$$ARG_{OPS_{SS}} : OPS \rightarrow S^*$$

bzw. die *Zielfunktion*

1) Vgl. BEIERLE ET AL. (1993), S. 192.

2) Die Typisierung von Relationssymbolen wird weiter unten vertieft.

3) Wenn aus dem Argumentationskontext hervorgeht, dass es sich um eine *Operationssymbol-*Typisierungsfunktion handelt, wird fortan auch verkürzt die Bezeichnung *Typisierungsfunktion* verwendet. Analog wird auch von einer Typisierungsfunktion gesprochen, wenn aus dm Argumentationskontext hervorgeht, dass es sich um eine Relationssymbol-Typisierungsfunktion handelt.

$$\mathrm{ZIEL}_{\mathrm{OPS}_{\mathrm{SS}}} : \mathrm{OPS} \rightarrow \mathrm{S}$$

verwendet. Sie wurden nicht explizit in der Definition sortierter Signaturen aufgeführt, da sich die Bildmengen beider Funktionen implizit aus der Typisierungsfunktion $\mathrm{typ}_{\mathrm{OPS}_{\mathrm{SS}}}$ ergeben. Für ein Operationssymbol $O_i \in \mathrm{OPS}$ mit $\mathrm{typ}_{\mathrm{OPS}_{\mathrm{SS}}}(O_i)=(s_1 \ldots s_n, s_{n+1})$ sind $\mathrm{ARG}_{\mathrm{OPS}_{\mathrm{SS}}}(O_i)=(s_1 \ldots s_n)$ und $\mathrm{ZIEL}_{\mathrm{OPS}_{\mathrm{SS}}}(O_i)=(s_{n+1})$.

Wenn ein Operationssymbol $O_i \in \mathrm{OPS}$ mit $\mathrm{typ}_{\mathrm{OPS}_{\mathrm{SS}}}(O_i)=(w,s)$, $w=\lambda$ und $s \in S$ gegeben ist, wird O_i als *Konstantensymbol zur Sorte s* bezeichnet. Jede sortenspezifische Menge KON_s mit $s \in S$ umfasst alle Konstantensymbole zu der Sorte s. Die Familie $\mathrm{KONF}_{\mathrm{SIG}_{\mathrm{SS}}}=(\mathrm{KON}_s)_{s \in S}$ umfasst wiederum alle sortenspezifischen Mengen von Konstantensymbolen über einer sortierten Signatur $\mathrm{SIG}_{\mathrm{SS}}$:

$$\mathrm{KONF}_{\mathrm{SIG}_{\mathrm{SS}}} = (\mathrm{KON}_s)_{s \in S}$$
$$\text{mit } \mathrm{KON}_s=\{O_i \mid O_i \in \mathrm{OPS} \wedge \mathrm{ARG}_{\mathrm{OPS}_{\mathrm{SS}}}(O_i)=\lambda\}.$$

Die Operationssymbole $O_1,\ldots,O_I \in \mathrm{OPS}$ werden im Rahmen der formalen Semantik extensional durch Operationen interpretiert. In Anlehnung hieran kann jedes Operationssymbol $O_i \in \mathrm{OPS}$ mit $\mathrm{typ}_{\mathrm{OPS}_{\mathrm{SS}}}(O_i)=(s_1 \ldots s_n, s_{n+1})$ durch eine funktionsnahe Notation der Form

$$O_i\colon s_1 \ldots s_n \rightarrow s_{n+1}$$

mit $n \in \mathbb{N}$ und $s_1,\ldots,s_n,s_{n+1} \in S$ angegeben werden.

Durch die funktionale Typisierung wird ein Formulierungsmittel ausgeschlossen, das insbesondere aus der objektorientierten Systemspezifikation bekannt ist. Es handelt sich dabei um die *polymorphe* Spezifikation von Operationssymbolen.[1)] Des Öfteren wird nämlich als zweite Komponente sortierter Signaturen statt der hier verwendeten Menge OPS eine Familie $\mathrm{OPSF}_{\mathrm{SIG}_{\mathrm{SS}}}=(\mathrm{OPS}_{w,s})_{w \in S^*, s \in S}$ *typspezifischer* Mengen von Operationssymbolen über einer sortierten Signatur $\mathrm{SIG}_{\mathrm{SS}}$ aufgeführt.[2)] Die Typisierungsfunktion $\mathrm{typ}_{\mathrm{OPS}_{\mathrm{SS}}}$ entfällt vollkommen. Der Typ jedes Operationssymbols O_i ergibt sich implizit aus den typspezifischen Mengen von Operationssymbolen, deren Element das Operationssymbol ist. Ein Operationssymbol O_i kann Element mehrerer unterschiedlicher typspezifischer Mengen von Operationssymbolen sein. Wenn ein Operationssymbol O_i Element von mindestens zwei unterschiedlichen typspezifischen Mengen OPS_{w_1,s_1} und OPS_{w_2,s_2} von Operationssymbolen ist, handelt es sich um ein polymorph spezifiziertes

1) Polymorphe Spezifikationen sind dem Verfasser lediglich im Zusammenhang mit Operationssymbolen bekannt. Die weitere Argumentation ließe sich allerdings ohne weiteres auch auf Relationssymbole übertragen.

2) Vgl. EHRICH ET AL. (1989), S. 14; EHRIG ET AL. (1999), S. 123; LOECKX ET AL. (1996), S. 26. Eine solche Charakterisierung ist dem Verfasser zwar nur für sortierte Signaturen bekannt. Unter Beibehaltung der formalen Vorgehensweise ist sie allerdings auch auf konventionelle Signaturen übertragbar.

Operationssymbol. Ein solches „Überladen“[1] von Operationssymboltypisierungen wird unter dem Begriff *Polymorphismus* weiter differenziert.[2]

Diese Funktionalitäten polymorpher Symbolspezifikation werden in der vorliegenden Arbeit durch Mechanismen gewährleistet, die einen Rückgriff auf Polymorphismus erübrigen. Zwar wird es auch im weiteren Verlauf notwendig, Terme in den Argumenten von Operationssymbolen aufzunehmen, die nicht (unmittelbar) zu den Sorten gehören, die für den Argumentbereich definiert sind. Allerdings wird diese Aufnahme „sortenfremder“ Terme nicht durch die polymorphe Spezifikation von Operationssymbolen erreicht, sondern durch die Erweiterung der Regeln zur Konstruktion sortenspezifischer Terme. Dieser Aspekt bildet ein wesentliches Charakteristikum *ontologischer Signaturen*, die in Kürze vorgestellt werden.[3]

Darüber hinaus könnte das Formulierungsmittel Polymorphismus auch bei einer funktionalen Typisierung von Operations- und Relationssymbolen beibehalten werden, wenn die Sorten einerseits aus dem Argument- und andererseits aus dem Zielbereich, mit denen ein Operationssymbol typisiert wird, vermengt würden. Hierzu müsste die Vorschrift für die *verkürzte Typisierungsfunktion* $typ_{OPS_{SS}}{}'$ lauten:

$$typ_{OPS_{SS}}{}': OPS \rightarrow S^+.$$

Der Typ zu einem Operationssymbol würde dann in der Form $(s_1...s_n s_{n+1})$ angegeben werden, wobei die Sortenkette $s_1...s_n$ den Argumentbereich und die letzte Sorte s_{n+1} den Zielbereich angeben. Das leere Wort λ würde als Sortenstruktur ausgeschlossen werden, da zumindest die Zielsorte angegeben werden muss.

Um eine polymorphe Spezifizierung von Operationssymbolen zuzulassen, würde die verkürzte Typisierungsfunktion $typ_{OPS_{SS}}{}'$ erneut zu

$$typ_{OPS_{SS}}{}'': OPS \rightarrow pot_+(S^+)$$

umformuliert werden. Durch die Potenzmenge $pot_+(S^+)$ der Menge S^+ aller nicht-leeren Sortenketten über S im Nachbereich der Typisierungsfunktion $typ_{OPS_{SS}}{}''$ werden für jedes Operationssymbol Funktionswerte mit mehreren nicht leeren Sortenketten erlaubt. Jede Sortenkette, die zu einem Operationssymbol zugeordnet ist, stellt in diesem Fall eine Typisierung des Operationssymbols dar. Allerdings ist bei dieser Art von Typisierung kein unmittelbarer formaler Zugriff auf den Argument- und den Zielbereich eines Operationssymbols möglich. Da dieser Aspekt für den weiteren Ausbau sortierter Signaturen zu ontologischen Signaturen von Vorteil ist, wird auf die o.a. Formulierung der Typisierungsfunktion auf Kosten polymorpher Ausdrucksmöglichkeiten verzichtet.

---

1) Vgl. EHRICH ET AL. (1989), S. 174; EHRIG ET AL. (1999), S. 124 („Overloading“).

2) Vgl. CARDELLI/WEGNER (1985), S. 475 ff.; GOGUEN/MESEGUER (1992), S. 219 ff.; MONIN (2003), S. 190; PAHL (1996), S. 81 ff.

3) Vgl. Abschnitt 3.1.2.1.

Die Menge $RS=\{R_1,...,R_J\}$ mit $j=1,...,J$ und $J\in\mathbb{N}$ umfasst Relationssymbole aus einer sortierten Signatur SIG_{SS}. Wenn die Menge RS der Relationssymbole leer ist, handelt es sich um eine *algebraische sortierte Signatur*. Bei einer nicht-leeren Menge RS aller Relationssymbole handelt es sich bei einer sortierten Signatur SIG_{SS} um eine *relationale sortierte Signatur*.

Jedem Relationssymbol $R_j\in RS$ werden durch die *Relationssymbol-Typisierungsfunktion*

$$typ_{RS_{SS}}: RS \rightarrow S^+$$

eine *Stelligkeit* und einer Sortenstruktur zugewiesen. Für ein Relationssymbol $R_j\in RS$ mit $typ_{RS_{SS}}(R_j)=s_1...s_n$ gibt die Relationssymbol-Typisierungsfunktion $typ_{RS_{SS}}$ die *Argumentsorten* an. Analog zu der Vorgehensweise bei Operationssymbolen kann auf die Sorten aus dem Argument eines Relationssymbols R_j durch die Argumentfunktion

$$ARG_{RS_{SS}}: RS \rightarrow S^+$$

zugegriffen werden. Zielsorten sind hingegen für Relationssymbole nicht vorgesehen. Da das Argument eines Relationssymbols $R_j\in RS_{SS}$ stets mit seiner Typisierung $typ_{RS_{SS}}(R_j)$ übereinstimmt, gilt für jedes Relationssymbol R_j:

$$ARG_{RS_{SS}}(R_j) = typ_{RS_{SS}}(R_j).$$

Für ein Relationssymbol $R_j\in RS$ mit $typ_{RS_{SS}}(R_j)=s_1...s_n$ gilt somit $ARG_{RS_{SS}}(R_j)=s_1...s_n$.

Die Mengenfamilie $VARF_{SIG_{SS}}=(VAR_s)_{s\in S}$ umfasst als Mitglieder sortenspezifische *Variablenmengen*. Die Elemente einer sortenspezifischen Variablenmenge VAR_s werden als *Variablen zur Sorte s* bezeichnet. Die Vereinigung aller sortenspezifischen Variablenmengen entspricht der Menge VAR aller Variablen aus dem formalsprachlichen Alphabet $ALPH_{SS}$:

$$VAR = \cup_{s\in S}VAR_s.$$

Für die Elemente der Menge VAR aller Variablen wird vorausgesetzt, dass sie sowohl zu der Menge S als auch zu den Mengen OPS und RS disjunkt ist:

$$(VAR \cap S) = (VAR \cap OPS) = (VAR \cap RS) = \varnothing.$$

Zudem werden die Mitglieder der Familie $VARF_{SIG_{SS}}$ aller sortenspezifischen Variablenmengen als paarweise disjunkt zueinander angenommen. Somit darf keine Variable in zwei unterschiedlichen sortenspezifischen Variablenmengen enthalten sein:

$$\forall s_1,s_2\in S: s_1\neq s_2 \rightarrow VAR_{s_1} \cap VAR_{s_2}=\varnothing.$$

Die Familie $VARF_{SIG_{SS}}$ aller sortenspezifischen Variablenmengen bildet gemeinsam mit der Familie $KONF_{SIG_{SS}}$ aller sortenspezifischen Mengen von Konstantensymbolen die Familie $INDF_{SIG_{SS}}$ aller sortenspezifischen Mengen von Individuensymbolen:

$$INDF_{SIG_{SS}}=(IND_s)_{s\in S}$$
$$\text{und } IND_s = VAR_s \cup KON_s.$$

Jede sortenspezifische Menge IND_s von Individuensymbolen setzt sich aus der entsprechenden sortenspezifischen Variablenmenge VAR_s und der ebenso entsprechenden sortenspezifischen Menge KON_s aller Konstantensymbole zusammen.

Zur Darstellung sortierter Signaturen wird die Darstellungsweise für konventionelle Signaturen in analoger Weise übernommen:

$\underline{SIG_{SS}}$

OPS:

O_1: $s_{O1_1} \ldots s_{O1_{l(O1)}} \rightarrow s_{O1_{l(O1+1)}}$

...

O_I: $s_{OI_1} \ldots s_{OI_{l(OI)}} \rightarrow s_{OI_{l(OI+1)}}$

RS:

R_1: $s_{R1_1} \ldots s_{R1_{l(R1)}}$

...

R_J: $s_{R1_J} \ldots s_{RJ_{l(RJ)}}$

$VARF_{SIG_{SS}}$:

$VAR_{s_1} = \{x_{1_1}, \ldots, x_{1_{l(s1)}}\}$

...

$VAR_{s_n} = \{x_{n_1}, \ldots, x_{n_{l(sn)}}\}$

2.2.1.2.2.2 Sortierte Ausdrücke

2.2.1.2.2.2.1 Sortierte Terme

Die Menge

$$EXPR_{SIG_{SS}} \subset ALPH_{SS}^*$$

umfasst alle *sortierten Ausdrücke* über einer sortierten Signatur SIG_{SS}. Sie setzt sich zum einen aus der Menge $TERM_{SIG_{SS}}$ aller Terme und zum anderen aus der Menge $FORM_{SIG_{SS}}$ aller Formeln über SIG_{SS} zusammen:

$$EXPR_{SIG_{SS}} = TERM_{SIG_{SS}} \cup FORM_{SIG_{SS}}$$
$$\text{mit } TERM_{SIG_{SS}} \cap FORM_{SIG_{SS}} = \varnothing.$$

Analog zu der Definition der Menge $FORM_{SIG_{KS}}$ aller Formeln über einer konventionellen Signatur SIG_{KS} werden bei der Definition der Menge $FORM_{SIG_{SS}}$ aller Formeln über einer sortierten Signatur SIG_{SS} Terme vorausgesetzt. Daher werden im Folgenden Terme über sortierten Signaturen vorgestellt, um darauf aufbauend Formeln über sortierten Signaturen vorstellen zu können.

Die Menge $TERM_{SIG_{SS}}$ aller Terme über einer sortierten Signatur SIG_{SS} wird *sortenspezifisch ausdifferenziert*. Diese geschieht, indem den Termen über einer sortierten Signatur SIG_{SS} *Sortenzugehörigkeiten* zugewiesen werden. Jede sortenspezifische Termmenge $TERM_s \subseteq TERM_{SIG_{SS}}$ umfasst sortierte Terme zu einer Sorte s. Die sortenspezifischen Termmengen, deren Elemente über einer sortierten Signatur SIG_{SS} konstruiert sind, werden zu der Familie $TERMF_{SIG_{SS}}$ aller sortenspezifischen Termmengen zusammengefasst. Dabei ist die Familie $TERMF_{SIG_{SS}}$ aller sortenspezifischen Termmengen als

$$TERMF_{SIG_{SS}} = (TERM_s)_{s \in S}$$

definiert.

Die Vereinigung aller sortenspezifischen Termmengen ist die Menge aller Terme, die über der Signatur SIG_{SS} konstruiert werden können:

$$TERM_{SIG_{SS}} = \cup_{s \in S} TERM_s.$$

Im Gegensatz zur Termmenge $TERM_{SIG_{SS}}$, in der alle Terme über der sortierten Signatur SIG_{SS} ohne Ausweis ihrer Sortenzugehörigkeiten undifferenziert enthalten sind, führt die Familie $TERMF_{SIG_{SS}}$ die selben Terme so auf, dass die Sortenzugehörigkeit der Terme aus der Zugehörigkeit zu genau einer der sortenspezifischen Termmengen $TERM_s$ unmittelbar erkannt werden kann.

Die sortenspezifischen Termmengen sind wie folgt induktiv definiert:

(1.) Wenn $x_q \in VAR_s$ gilt, dann gilt $x_q \in TERM_s$. Jede Variable x_q aus einer sortenspezifischen Variablenmenge VAR_s ist ein *atomarer sortierter Term* zur Sorte s.

(2.) Wenn $O_i \in OPS$ ein null-stelliges Operationssymbol mit $typ_{OPS_{SS}}(O_i)=(\lambda,s)$ und infolgedessen mit der Zielsorte s ist, dann gilt $O_i \in TERM_s$. Das Operationssymbol O_i ist ein *atomarer sortierter Term* zur Sorte s.

(3.) Wenn $O_i \in OPS$ ein Operationssymbol mit $typ_{OPS_{SS}}(O_i)=(s_1...s_n,s_{n+1})$, $n \in \mathbb{N}_+$ mit $s_1,...,s_n,s_{n+1} \in S$ und infolgedessen mit der Zielsorte s_{n+1} ist und wenn $t_1,...,t_n$ jeweils sortierte Terme mit $t_x \in TERM_{s_x}$ mit x=1,...,n und $n \in \mathbb{N}_+$ sind, dann ist $O_i(t_1,...,t_n)$ ein *zusammengesetzter sortierter Term* zur Sorte s_{n+1}.[1)]

Wie bei der Definition der Menge $TERM_{SIG_{KS}}$ aller Terme über konventionellen Signaturen auch, finden Relationssymbole bei der Definition der Familie $TERMF_{SIG_{SS}}$ aller

1) Bei der Typisierung von Operations- und Relationssymbolen wird eine vereinfachte Schreibweise übernommen, um die Sortenstruktur nicht zu verkomplizieren. Bei einer präzisieren Schreibweise müsste der Typ $typ_{OPS_{SS}}(O_i)$ eines Operationssymbols O_i in der Form

$$typ_{OPS_{SS}}(O_i) = (s_{1(i)}...s_{n(i)},s_{n+1(i)})$$

angegeben werden. Damit die Zeichenkette $O_i(t_1,...,t_n)$ einen zulässigen Term darstellt, müsste zudem $t_x \in TERM_{s_{n(x)}}$ für x=1,...,n gelten. Auch eine derartige Schreibweise kann jedoch nicht zufrieden stellen, weil die Länge n jedes Typs symbolspezifisch variieren kann. Daher wird die gängige Schreibweise übernommen und darauf hingewiesen, dass es sich bei den Sorten aus dem Typ $(s_1...s_n,s_{n+1})$ nicht um die Sorten mit der gleichen Bezeichnung aus der Sortenmenge S handeln muss.

sortenspezifischen Termmengen keine Berücksichtigung. Sortierte Terme gehen entweder aus sortenspezifischen Variablen, aus Konstantensymbolen oder aus der Anwendung von Operationssymbolen auf deren Argumente hervor.

Entsprechend der o.a. Regel (1.) ist jede Variable aus einer sortenspezifischen Variablenmenge VAR_s ein Term zu der sortenspezifischen Termmenge $TERM_s$. Somit sind die Mitglieder der Familie $VARF_{SIG_{SS}}$ jeweils Teilmengen der Mitglieder der Familie $TERMF_{SIG_{SS}}$:

$$\forall s \in S: VAR_s \subseteq TERM_s.$$ [1)]

Sortierte Terme, die aus Operationssymbolen hervorgehen, können in zwei Unterfälle unterschieden werden. Der erste Unterfall umfasst solche Terme, die aus Konstantensymbolen hervorgehen. Bei Konstantensymbolen aus einer sortierten Signatur SIG_{SS} handelt es sich um Operationssymbole mit einem leeren Argumentbereich. Um den Typ von Termen zu bestimmen, die aus Operationssymbolen mit leerem Argumentbereich hervorgehen, ist die Zielsorte von Bedeutung. Jedes Konstantensymbol ist nämlich ein Term zur Zielsorte des null-stelligen Operationssymbols, aus dem es hervorgeht. Somit sind die Mitglieder der Familie $KONF_{SIG_{SS}}$ jeweils Teilmengen der Mitglieder der Familie $TERMF_{SIG_{SS}}$:

$$\forall s \in S: KON_s \subseteq TERM_s.$$

Eine sortenspezifische Menge IND_s von Individuensymbolen ist somit immer eine Teilmenge der entsprechenden sortenspezifischen Termmenge $TERM_s$:[2)]

$$\forall s \in S: IND_s \subseteq TERM_s.$$

Der zweite Unterfall umfasst solche Terme, die aus der Anwendung von Operationssymbolen auf andere Terme hervorgehen. Während bei der Definition von zusammengesetzten konventionellen Termen lediglich die *Anzahl* der konventionellen Terme im Argument von Operationssymbolen berücksichtigt wurde, wird bei der Definition von zusammengesetzten sortierten Termen auch darauf geachtet, welche *Sortenzugehörigkeit* die Terme im Argument von Operationssymbolen haben. Es dürfen an den jeweili-

---

1) Zu beachten ist, dass hingegen $VARF_{SIG_{SS}} \subseteq TERMF_{SIG_{SS}}$ nicht gelten muss. Jedes Mitglied $TERM_s$ der Familie $TERMF_{SIG_{SS}}$ umfasst Terme zur entsprechenden Sorte s. In einer sortenspezifischen Termmenge $TERM_s$ sind somit auch alle Variablen zur Sorte s enthalten. Darüber hinaus können allerdings in der sortenspezifischen Termmenge $TERM_s$ auch solche Terme enthalten sein, die unmittelbar aus Operationssymbolen (Konstantensymbolen) oder aus der typgerechten Anwendung von Operationssymbolen auf sortierte Terme hervorgehen.

2) Zu beachten ist, dass nicht notwendigerweise

$$INDF_{SIG_{SS}} \subseteq TERMF_{SIG_{SS}}$$

gelten muss. In einer sortenspezifischen Termmenge $TERM_s$ können auch zusammengesetzte sortierte Terme vorkommen. In diesem Fall würde es in $TERMF_{SIG_{SS}}$ mindestens eine sortenspezifische Termmenge $TERM_s$ geben, in der über die Elemente der entsprechenden sortenspezifischen Menge von Individuensymbolen hinaus noch zusammengesetzte sortierte Terme existieren würden.

gen Argumentstellen nämlich nur solche sortierten Terme verwendet werden, die zu derjenigen Sorte gehören, die durch die Typisierungsfunktion $typ_{OPS_{SS}}$ für die jeweils betroffene Argumentstelle des Operationssymbols vorgeschrieben wird.

Die Klassifikation von konventionellen Termen lässt sich analog auf sortierte Terme übertragen. Demnach können auch sortierte Terme hinsichtlich ihrer *Zusammengesetztheit* und ihres *Variablenanteils* klassifiziert werden.

Hinsichtlich ihrer Zusammengesetztheit können sortierte Terme in die zueinander disjunkten Mengenfamilien $ATF_{SIG_{SS}}=(AT_s)_{s\in S}$ und $ZTF_{SIG_{SS}}=(ZT_s)_{s\in S}$ eingeordnet werden. Die Mengenfamilie $ATF_{SIG_{SS}}$ umfasst sortenspezifische Mengen *atomarer sortierter Terme*:

$$ATF_{SIG_{SS}} = (AT_s)_{s\in S}$$
$$\text{mit } AT_s = \{t \mid t\in TERM_s \land t\in IND_s\}$$
$$\text{und } AT_{SIG_{SS}} = \cup_{s\in S} AT_s.$$

Die Mengenfamilie $ZTF_{SIG_{SS}}$ umfasst sortenspezifische Mengen zusammengesetzter sortierter Terme:

$$ZTF_{SIG_{SS}} = (ZT_s)_{s\in S}$$
$$\text{mit } ZT_s = \{t \mid t\in TERM_s \land t\notin IND_s\}$$
$$\text{und } ZT_{SIG_{SS}}=\cup_{s\in S}ZT_s$$

Hinsichtlich des Variablenanteils können Terme über einer sortierten Signatur SIG_{SS} in die zueinander disjunkten Mengenfamilien $VTF_{SIG_{SS}}$ und $GTF_{SIG_{SS}}$ eingeordnet werden. Die Familie $VTF_{SIG_{SS}}$ umfasst sortenspezifische Mengen *variabler Terme*:

$$VTF_{SIG_{SS}}=(VT_s)_{s\in S}.$$

Die Familie $GTF_{SIG_{SS}}$ umfasst sortenspezifische Mengen von *Grundtermen*:

$$GTF_{SIG_{SS}}=(GT_s)_{s\in S}.$$

Jede sortenspezifische Termmenge $TERM_s$ setzt sich aus einer sortenspezifischen Menge VT_s variabler Terme und einer sortenspezifischen Menge GT_s von Grundtermen zusammen:

$$\forall s\in S\text{: } TERM_s = VT_s \cup GT_s.$$

Die Bestimmung des Variablenanteils von sortierten Termen erfolgt durch die Funktion

$$var_{T_{SS}} : TERM_{SIG_{SS}} \rightarrow pot(VAR).$$

Die Bilder der Funktion $var_{T_{KS}}$ lassen sich auf die Funktion $var_{T_{SS}}$ übertragen.[1] Die Funktionen $var_{T_{SS}}$ ordnet jedem sortierten Term $t\in TERM_{SIG_{SS}}$ die Menge von Variablen zu, die im Term t vorkommen. Dabei sind die Bilder für die Variablenfunktion $var_{T_{SS}}$ wie folgt definiert:

1) Vgl. Abschnitt 2.2.1.1.2.2.1.

(1.) $var_{T_{SS}}(x_q) = \{x_q\}$
für $x_q \in VAR$,

(2.) $var_{T_{SS}}(O_i) = \varnothing$
für $O_i \in OPS$ mit $ARG_{OPS_{SS}}(O_i)=\lambda$ und

(3.) $var_{T_{SS}}(O_i(t_1,...,t_n)) = var_{T_{SS}}(t_1) \cup ... \cup var_{T_{SS}}(t_n)$
für $O_i \in OPS$ mit $typ_{OPS_{SS}}(O_i)=(s_1...s_n,s_{n+1})$.

Für einen Term $t \in TERM_{SIG_{SS}} \cap VAR$, der eine Variable x_q ist, beinhaltet die Menge $var_{T_{SS}}(t)$ gemäß der Regel (1.) nur die Variable x_q. Sortierte Terme, die aus Konstantensymbolen hervorgehen, weisen hingegen gemäß der Regel (2.) grundsätzlich keinen Variablenanteil auf. Der Variablenanteil $var_{T_{SS}}(t)$ eines zusammengesetzten sortierten Terms $t \in ZT_{SIG_{SS}}$ stimmt gemäß der Regel (3.) mit der Vereinigungsmenge der Variablenanteile überein, die die sortierten Terme $t_1,...,t_n$ im Argument des erstgenannten Terms t haben. Dabei erlaubt das Rekursionsschema für die Variablenfunktion $var_{T_{SS}}$ auch die Bestimmung der Variablenanteile von solchen zusammengesetzten sortierten Termen, die in ihren Argumenten selbst zusammengesetzte sortierte Terme aufweisen.

Jede sortenspezifische Menge

$$VT_s=\{t \mid t \in TERM_s \wedge var_{T_{SS}}(t) \neq \varnothing\}$$

umfasst alle *variablen sortierten Terme* zur Sorte s. Für die Mengen VAR_s s-spezifischer Variablen gilt hierbei, dass sie Teilmengen der s-spezifischen Menge VT_s variabler sortierter Terme sind:

$$VAR_s \subseteq VT_s.$$

Jede sortenspezifische Menge

$$GT_s=\{t \mid t \in TERM_s \wedge var_{T_{SS}}(t) = \varnothing\}$$

umfasst *sortierte Grundterme*. Die sortenspezifischen Mengen von Konstantensymbolen sind jeweils Teilmengen der sortenspezifischen Mengen von Grundtermen:

$$KON_s \subseteq GT_s.$$

2.2.1.2.2.2.2 Sortierte Formeln

Die zweite Teilmenge $FORM_{SIG_{SS}}$ der Menge $EXPR_{SIG_{SS}}$ aller Ausdrücke über einer sortierten Signatur SIG_{SS} umfasst *Formeln*. Um Formeln, die über sortierten Signaturen SIG_{SS} konstruiert werden, von Formeln, die über konventionellen Signaturen SIG_{KS} konstruiert werden abzugrenzen, werden Erstgenannte im Folgenden als *sortierte Formeln* bezeichnet, wenn aus dem Argumentationskontext nicht ersichtlich ist, um welche Art von Formeln es sich handelt.

Die induktive Grammatik zur Konstruktion sortierter Formeln lautet:

(1.) Es gilt $w,f \in FORM_{SIG_{SS}}$. w ist die immer gültige, tautologische Formel; f ist die immer ungültige, kontradiktorische Formel.

(2.) Wenn $R_j \in RS$, $typ_{RS_{SS}}(R_j)=(s_1...s_n)$ und $t_1...t_n \in TERM_{SIG_{SS}}$ mit $t_x \in TERM_{s_x}$ mit $x=1,...,n$ gelten, dann gilt auch $R_j(t_1,...,t_n) \in FORM_{SIG_{SS}}$. Bei dem Ausdruck $R_j(t_1,...,t_n)$ handelt es sich um eine *atomare sortierte Formel*.

(3.) Wenn $F \in FORM_{SIG_{SS}}$ gilt, dann gilt auch $\neg F \in FORM_{SIG_{SS}}$. Bei dem Ausdruck $\neg F$ handelt es sich um eine *zusammengesetzte sortierte Formel*.

(4.) Wenn $F_1,F_2 \in FORM_{SIG_{SS}}$ gilt, dann gelten auch $F_1 \wedge F_2$, $F_1 \vee F_2$, $F_1 \underline{\vee} F_2$, $F_1 \rightarrow F_2$, $F_1 \leftrightarrow F_2 \in FORM_{SIG_{SS}}$. Es handelt sich hierbei um *zusammengesetzte sortierte Formeln*.

(5.) Wenn x eine Variable ist und F eine sortierte Formel ist, dann gelten $\forall x{:}F$, $\exists x{:}F$, $\underline{\exists} x{:}F \in FORM_{SIG_{SS}}$. Es handelt sich hierbei um *zusammengesetzte sortierte Formeln*.

Die Induktionsbasis der Grammatik zur Konstruktion sortierter Formeln sind die Regeln (1.) und (2.) zur Konstruktion atomarer sortierter Formeln. Eine atomare sortierte Formel geht entweder mit Hilfe von Regel (1.) aus w bzw. f oder mit Hilfe von Regel (2.) aus der typgerechten Anwendung eines Relationssymbols R_j auf sortierte Terme hervor. Atomare sortierte Formeln können demnach konstruiert werden, indem ein Relationssymbol R_j mit der Typisierung $typ_{RS_{SS}}(R_j)=(s_1...s_n)$ auf n Terme angewendet wird. Bis zu diesem Punkt stimmen die Induktionsbasen der Grammatiken zur Konstruktion konventioneller und sortierter Signaturen überein. Der Unterschied zwischen den Grammatiken liegt darin, dass in der Grammatik zur Konstruktion sortierter Formeln zusätzlich gefordert wird, dass die Terme $t_1,...,t_n$ im Argument einer atomaren sortierten Formel $R_j(t_1,...,t_n)$ auch den Sorten $s_1,...,s_n$ zugewiesen sein müssen, mit denen die jeweilige Argumentstelle des Relationssymbols R_j typisiert ist. In diesem Punkt unterscheiden sich die Grammatiken zur Konstruktion atomarer konventioneller und atomarer sortierter Formeln. Entsprechend unterscheiden sich zusammengesetzte konventionelle Formeln von zusammengesetzten sortierten Formeln dadurch, dass es sich bei den Formelteilen der letztgenannten Art um sortierte Formeln handelt.

Für die Klassifikation sortierter Formeln gelten Bedingungen, die bereits für konventionelle Formeln aufgeführt wurden, analog.[1] Demnach kann die Menge $FORM_{SIG_{SS}}$ aller sortierten Formeln über einer sortierten Signatur SIG_{SS} in die Teilmenge AF_{SS} aller atomaren sortierten Formeln und die Teilmenge ZF_{SS} aller zusammengesetzten sortierten Formeln unterteilt werden:

$$FORM_{SIG_{SS}} = AF_{SIG_{SS}} \cup ZF_{SIG_{SS}}$$
$$\text{mit } AF_{SIG_{SS}} \cap ZF_{SIG_{SS}} = \varnothing.$$

Die Unterteilung erfolgt mittels der Teilformelfunktion

$$tf_{SS}: FORM_{SIG_{SS}} \rightarrow pot_{+}(FORM_{SIG_{SS}}).$$

1) Vgl. Abschnitt 2.2.1.1.2.2.2.

Die Funktion tf_{SS} weist jeder sortierten Formel $F \in FORM_{SIG_{SS}}$ die Menge von sortierten Formeln zu, deren Elemente die ursprüngliche sortierte Formel $F \in FORM_{SIG_{SS}}$ und sich selbst als Teilformeln hat:

(1.) $tf_{SS}(w) \quad =\{w\}$,

(2.) $tf_{SS}(f) \quad =\{f\}$,

(3.) $tf_{SS}(R_j(t_1,...,t_m)) \quad =\{R_j(t_1,...,t_m)\}$,

(4.) $tf_{SS}(\neg F) \quad =\{\neg F\} \cup tf_{SS}(F)$,

(5.) $tf_{SS}(F_1 \bullet F_2) \quad =\{(F_1 \bullet F_2)\} \cup tf_{SS}(F_1) \cup tf_{SS}(F_2)$ für $\bullet \in \{\wedge, \vee, \underline{\vee}, \rightarrow, \leftrightarrow\}$ und

(6.) $tf_{SS}(\bullet x{:}F) \quad =\{\bullet x{:}F\} \cup tf_{SS}(F)$ für $\bullet \in \{\forall, \exists, \underline{\exists}\}$.

Die Mengen $AF_{SIG_{SS}}$ und $ZF_{SIG_{SS}}$ lassen sich mit Hilfe der Funktion tf_{SS} bestimmen als:

$$AF_{SIG_{SS}} = \{F \mid F \in FORM_{SIG_{SS}} \wedge tf_{SS}(F) = \{F\}\}$$
$$\text{und } ZF_{SIG_{SS}} = \{F \mid F \in FORM_{SIG_{SS}} \wedge tf_{SS}(F) \supset \{F\}).$$

Die Menge $AF_{SIG_{SS}}$ umfasst alle atomaren Formeln über einer sortierten Signatur SIG_{SS}. Entweder handelt es sich hierbei um solche atomare sortierte Formeln, die unmittelbar aus w oder f hervorgehen, oder um atomare sortierte Formeln, die aus der typgerechten Anwendung eines Relationssymbols R_j mit $typ_{RS_{SS}}(R_j)=(s_1...s_n)$ auf n Terme hervorgehen. Die Bilder atomarer sortierter Formeln entsprechend der Teilformelfunktion tf_{SS} stimmen mit ihnen selbst überein. Die Bilder zusammengesetzter sortierter Formeln enthalten hingegen mehrere sortierte Formeln. Ihrem Rekursionsschema folgend sind in den Bildern zu zusammengesetzten sortierten Formeln alle ihre Teilformeln und sie selbst enthalten.

Hinsichtlich ihres Variablenanteils kann die Menge $FORM_{SIG_{SS}}$ aller sortierter Formeln in die Mengen $GF_{SIG_{SS}}$ aller sortierten Grundformeln und die Menge $VF_{SIG_{SS}}$ aller sortierten variablen Formeln unterteilt werden:

$$FORM_{SIG_{SS}}=GF_{SIG_{SS}} \cup VF_{SIG_{SS}}$$
$$\text{mit } GF_{SIG_{SS}} \cap VF_{SIG_{SS}}=\varnothing.$$

Der Variablenanteil einer sortierten Formel wird über die Funktion

$$var_{F_{SS}}: FORM_{SIG_{SS}} \rightarrow pot(VAR)$$

bestimmt. Die Funktion $var_{F_{SS}}$ ordnet jeder sortierten Formel über einer sortierten Signatur SIG_{SS} die Menge der Variablen zu, die in ihr enthalten ist. Sie ist analog zu der Funktion $var_{F_{KS}}$ definiert, die den Anteil an Variablen von Formeln über einer konventionellen Signatur SIG_{KS} bestimmt.

(1.) $var_{F_{SS}}(w) \quad = \varnothing$,

(2.) $var_{F_{SS}}(f) \quad = \varnothing$,

(3.) $var_{F_{SS}}(R_j(t_1,...,t_n)) \quad = var_{T_{SS}}(t_1) \cup ... \cup var_{T_{SS}}(t_n)$
für alle $R_j \in RS_{SS}$ mit $typ_{RS_{SS}}(R_j)=(s_1...s_n)$, $t_x \in TERM_{s_x}$ und $x=1,...,n$,

(4.) $var_{F_{SS}}(\neg F) = var_{F_{SS}}(F)$
für alle $F \in FORM_{SIG_{SS}}$,

(5.) $var_{F_{SS}}(F_1 \bullet F_2) = var_{F_{SS}}(F_1) \cup var_{F_{SS}}(F_2)$
für alle $F_1, F_2 \in FORM_{SIG_{SS}}$ und $\bullet \in \{\wedge, \vee, \underline{\vee}, \rightarrow, \leftrightarrow\}$,

(6.) $var_{F_{SS}}(\bullet{:}F) = var_{F_{SS}}(F) \cup \{x\}$
für alle $F \in FORM_{SIG_{SS}}$ und $\bullet \in \{\forall x, \exists x, \underline{\exists} x\}$.

Die Funktion $var_{F_{SS}}$ ordnet jeder sortierten Formel $F \in FORM_{SIG_{SS}}$ die Menge an Variablen zu, die in der Formel F enthalten sind. Für die atomaren Formeln w und f ist ihr Variablenanteil die leere Menge $\varnothing$.

Die Mengen $GF_{SIG_{SS}}$ bzw. $VF_{SIG_{SS}}$ werden mittels der Variablenfunktion $var_{F_{SS}}$ wie folgt definiert:

$$GF_{SIG_{SS}} = \{F \mid F \in FORM_{SIG_{SS}} \wedge var_{F_{SS}}(F) = \varnothing\}$$
$$\text{und } VF_{SIG_{SS}} = \{F \mid F \in FORM_{SIG_{SS}} \wedge var_{F_{SS}}(F) \supset \varnothing\}$$

Hinsichtlich des Anteils gebundener Variablen lässt sich die Menge $FORM_{SIG_{SS}}$ aller sortierten Formeln in die Mengen $OF_{SIG_{SS}}$ aller *offenen sortierten Formeln* und die Menge $CF_{SIG_{SS}}$ aller *geschlossenen sortierten Formeln* unterteilen:

$$FORM_{SIG_{SS}} = OF_{SIG_{SS}} \cup CF_{SIG_{SS}}$$
$$\text{mit } OF_{SIG_{SS}} \cap CF_{SIG_{SS}} = \varnothing.$$

Der Anteil freier Variablen in einer sortierten Formel wird über die Funktion

$$fvar_{SS}: FORM_{SIG_{SS}} \rightarrow pot(VAR)$$

bestimmt. Sie gibt den Anteil $fvar_{SS}(F)$ freier Variablen in einer sortierten Formel $F \in FORM_{SIG_{SS}}$ wie folgt wieder:

(1.) $fvar_{SS}(w) = \varnothing$,

(2.) $fvar_{SS}(f) = \varnothing$,

(3.) $fvar_{SS}(R_j(t_1,...,t_n)) = var_{T_{SS}}(t_1) \cup ... \cup var_{T_{SS}}(t_n)$
für alle $R_j \in RS$ mit $typ_{RS_{SS}}(R_j) = (s_1...s_n)$ und $t_x \in TERM_{s_x}$ für alle $x=1,...,n$,

(4.) $fvar_{SS}(\neg F) = fvar_{SS}(F)$
für alle $F \in FORM_{SIG_{SS}}$,

(5.) $fvar_{SS}(F_1 \bullet F_2) = fvar_{SS}(F_1) \cup fvar_{SS}(F_2)$
für alle $F_1, F_2 \in FORM_{SIG_{SS}}$ und $\bullet \in \{\wedge, \vee, \underline{\vee}, \rightarrow, \leftrightarrow\}$ und

(6.) $fvar_{SS}(\bullet F) = fvar_{SS}(F) \backslash \{x\}$
für alle $F \in FORM_{SIG_{SS}}$ und $\bullet \in \{\forall x, \exists x, \underline{\exists} x\}$.

Die Mengen $OF_{SIG_{SS}}$ bzw. $CF_{SIG_{SS}}$ werden mit Hilfe der Funktion $fvar_{SS}$ wie folgt definiert:

$$OF_{SIG_{SS}} = \{F \mid F \in FORM_{SIG_{SS}} \wedge fvar_{SS}(F) \supset \varnothing\}$$
$$\text{und } CF_{SIG_{SS}} = \{F \mid F \in FORM_{SIG_{SS}} \wedge fvar_{SS}(F) = \varnothing\}.$$

2.2.1.2.3 Semantische Aspekte der sortierten Prädikatenlogik

2.2.1.2.3.1 SIG_{SS}-Strukturen

Die extensionale Interpretation einer sortierten Signatur SIG_{SS} erfolgt durch eine SIG_{SS}-Struktur $A_{SIG_{SS}}$. Die Komponenten einer SIG_{SS}-Struktur $A_{SIG_{SS}}$ sind Extensionen von Komponenten der sortierten Signatur SIG_{SS}.

Es handelt sich hierbei erstens um eine Familie OBF_{SS} von *sortenspezifischen Objektmengen*. Durch die sortenspezifischen Objektmengen $OB_{s_1},...,OB_{s_n} \in OBF_{SS}$ werden die Sorten $s_1,...,s_n \in S$ der sortierten Signatur SIG_{SS} extensional interpretiert. Jedes formale Objekt $ob_u \in OB_s$ aus der Extension OB_s einer Sorte s ist eine Instanz der Sorte s.

Als zweite Komponente umfasst eine SIG_{SS}-Struktur $A_{SIG_{SS}}$ eine Familie OPF von *Operationen*. Durch jede Operation $o_i \in OPF$ wird genau ein Operationssymbol $O_i \in OPS$ aus der sortierten Signatur SIG_{SS} extensional interpretiert.

Als drittes umfasst eine SIG_{SS}-Struktur $A_{SIG_{SS}}$ eine Familie RF von *Relationen*. Jedes Relationssymbol $R_1,...,R_j \in RS$ aus einer sortierten Signatur SIG_{SS} wird durch genau eine Relation $r_j \in RF$ extensional interpretiert.

Die Abbildungen der Sorten und Operations- und Relationssymbole auf sortenspezifische Objektmengen, Operationen bzw. Relationen erfolgt durch die Interpretationsfunktionen I_S, I_{OPS} bzw. I_{RS}.

Eine SIG_{SS}-Struktur $A_{SIG_{SS}}$ zu einer sortierten Signatur SIG_{SS} ist definiert als:

$$A_{SIG_{SS}}=(OBF_{SS},OPF,RF,IF_{SS}).$$

Die Komponenten einer SIG_{SS}-Struktur $A_{SIG_{SS}}$ sind:

(1.) Eine Familie $OBF_{SS}=(OB_s)_{s \in S}$ von *sortenspezifischen Objektmengen*,

(2.) eine Familie $OPF=(o_1,...,o_I)$ von *Operationen* o_i mit $i=1,...,I$ und $I \in \mathbb{N}$,

(3.) eine Familie $RF=(r_1,...,r_J)$ von *Relationen* r_j mit $j=1...J$ und $J \in \mathbb{N}$ und,

(4.) eine Familie $IF_{SS}=(I_S,I_{OPS},I_{RS})$ von *Interpretationsfunktionen*.

Die Verknüpfung einer sortierten Signatur SIG_{SS} mit einer sortierten Struktur $A_{SIG_{SS}}$ erfolgt durch eine Familie $IF_{SS}=(I_S,I_{OPS},I_{RS})$ von *Interpretationsfunktionen* mit

$$I_S: S \rightarrow OBF_{SS}$$
$$I_{OPS}: OPS \rightarrow OPF$$
$$\text{und } I_{RS}: RS \rightarrow RF.$$

Bei den Interpretationsfunktionen I_S, I_{OPS} und I_{RS} handelt es sich um *linkstotale*, *bijektive* Funktionen. Sie sind einerseits linkstotal, da sie *alle* Elemente aus ihrem jeweiligen Argumentbereich auf ein Element des entsprechenden Zielbereichs abbilden. Dadurch wird gewährleistet, dass eine SIG_{SS}-Struktur $A_{SIG_{SS}}$ für *alle* Komponenten einer sortierten Signatur SIG_{SS} eine extensionale Interpretation bereithält. Würden auch partielle Interpretationsfunktionen zugelassen werden, wären auch solche SIG_{SS}-Strukturen erlaubt, die für einzelne Komponenten einer sortierten Signatur SIG_{SS} keine Extension bereithalten.

Andererseits handelt es sich bei den Interpretationsfunktionen I_S, I_{OPS} und I_{RS} um *bijektive* Funktionen. Auf die Bijektivität der Interpretationsfunktionen I_{OPS} und I_{RS} wurde bereits im Kontext der konventionellen Prädikatenlogik hingewiesen.[1)] Zum einen sind die Interpretationsfunktionen nämlich *injektiv*. Das heißt, dass die Elemente aus dem Zielbereich einer Interpretationsfunktion zu *höchstens einer* Komponente einer sortierten Signatur SIG_{SS} die entsprechende Extension sind. Dadurch wird gewährleistet, dass keine zwei Konstrukte aus einer sortierten Signatur SIG_{SS} durch das gleiche Konstrukt aus einer SIG_{SS}-Struktur $A_{SIG_{SS}}$ extensional interpretiert werden können. Zum anderen handelt es sich bei den Interpretationsfunktionen um *surjektive* Funktionen. *Jede* Komponente einer SIG_{SS}-Struktur $A_{SIG_{SS}}$ wird nämlich für die extensionale Interpretation *mindestens* einer Komponente einer sortierten Signatur SIG_{SS} verwendet. Somit interpretiert jede Komponente einer SIG_{SS}-Struktur $A_{SIG_{SS}}$ genau eine Komponente der sortierten Signatur SIG_{SS} extensional.

Die Interpretationsfunktion I_S bildet jede Sorte $s \in S$ auf eine Objektmenge $I_S(s)=OB_s$ aus der Familie OBF_{SS} aller sortenspezifischen Objektmengen ab. Dadurch wird jede Sorte $s \in S$ durch eine sortenspezifische Objektmenge $I_S(s)=OB_s$ interpretiert. Die formalen Objekte aus einer sortenspezifischen Objektmenge OB_s sind *Individuen* zur Sorte s.

Mit der Interpretationsfunktion I_{OPS} wird jedes Operationssymbol

$$O_i \in OPS$$
$$\text{mit } typ_{OPS_{SS}}(O_i)=(s_1 \ldots s_n, s_{n+1})$$

durch eine Operation

$$I_{OPS}(O_i)=o_i$$
$$\text{mit } o_i\colon OB_{s_1} \times \ldots \times OB_{s_n} \rightarrow OB_{s_{n+1}}$$

extensional interpretiert.

Jedes Relationssymbol

$$R_j \in RS$$
$$\text{mit } typ_{RS_{SS}}(R_j)=(s_1 \ldots s_n)$$

wird durch eine Relation

$$I_{RS}(R_j)=r_j$$
$$\text{mit } r_j \subseteq OB_{s_1} \times \ldots \times OB_{s_n}$$

extensional interpretiert.

SIG_{SS}-Strukturen unterscheiden sich von SIG_{KS}-Strukturen fundamental in ihrer ersten Komponente. Während für alle SIG_{KS}-Strukturen ein unsortierter Objektbereich OB festgelegt wird, wird in SIG_{SS}-Strukturen eine Familie OBF_{SS} sortenspezifischer Ob-

1) Vgl. Abschnitt 2.2.1.1.3.1.

jektmengen vorausgesetzt. Jede sortenspezifische Objektmenge OB_s umfasst Individuen der Sorte s. Dadurch erlauben es SIG_{SS}-Strukturen, Individuen hinsichtlich ihrer „Art“ zu differenzieren. In der konventionellen Prädikatenlogik wird lediglich die Spezifikation von Individuen als Elemente der undifferenzierten Objektmenge OB zugelassen.

Das prädikatenlogische Universum OB geht aus der Vereinigung

$$OB = \cup_{s \in S} OB_s$$

aller sortenspezifischen Objektmengen hervor. Dabei hat nicht zu gelten, dass die sortenspezifischen Objektmengen zueinander disjunkt sein müssen. Die „Überlappung“ sortenspezifischer Objektmengen ist durchaus erlaubt.[1)]

Wenn in der sortierten Signatur SIG_{SS} nur eine Sorte $s \in S$ enthalten ist, stimmt die Menge OB mit der einzigen sortenspezifischen Objektmenge OB_s überein. In diesem Fall kann die SIG_{SS}-Struktur $A_{SIG_{SS}}$ aus zwei Perspektiven charakterisiert werden. In ihrer originären Charakterisierung handelt es sich um eine SIG_{SS}-Struktur im Sinn der sortierten Prädikatenlogik. Die Mengenfamilie OBF_{SS} hat dann nur ein Mitglied, nämlich die Objektmenge OB_s zu der einzigen Sorte s. Aus der zweiten Perspektive stimmt die SIG_{SS}-Struktur mit einer SIG_{KS}-Struktur $A_{SIG_{KS}}$ überein. Da die Mengenfamilie OBF_{SS} in diesem Fall nur eine sortenspezifische Objektmenge OB_s als Mitglied hat, kann statt der Mengenfamilie OBF_{SS} auch die undifferenzierte Objektmenge OB verwendet werden. Die SIG_{SS}-Struktur $A_{SIG_{SS}}$ degeneriert in diesem Fall zu einer SIG_{KS}-Struktur $A_{SIG_{KS}}$.

Die Individuen aus einer SIG_{SS}-Struktur $A_{SIG_{SS}}$ können auch eine *innere Struktur* aufweisen.[2)] Diese Alternative ergibt sich aus der Möglichkeit, Operationssymbole aus einer sortierten Signatur SIG_{SS} mit einer Konkatenation von Sorten zu typisieren. Die Objektmengen zu den Zielsorten solcher Operationssymbole umfassen dann formale Objekte, die aus der Anwendung von Operationen auf andere formale Objekte hervorgehen. Es handelt sich bei den letztgenannten formalen Objekten um Individuen aus Objektmengen zu solchen Sorten, die mit den Sorten im Argumentbereich des Operationssymbols übereinstimmen. Wenn z.B. ein Operationssymbol $O_1 \in OPS$ mit $typ_{OPS_{SS}}(O_1)=(s_{1_1}...s_{1_n},s_{n+1})$ in der Signatur SIG_{SS} spezifiziert ist, dann kann die Objektmenge $OB_{s_{n+1}}$ aus einer SIG_{SS}-Struktur $A_{SIG_{SS}}$ formale Objekte der Form $ob_1=o_1(ob_{1_1},...,ob_{1_n})$ umfassen, wobei die formalen Objekte $ob_{1_1},...,ob_{1_n}$ den sortenspezifischen Objektmengen $OB_{s_{1.1}},...,OB_{s_{1.n}}$ entstammen müssen. Jedes der formalen Objekte $ob_{1_1},...,ob_{1_n}$ kann wiederum selbst eine innere Struktur aufweisen, wenn die entsprechende Sorte im Argumentbereich des Operationssymbols als Zielsorte eines anderen Operationssymbols definiert ist.

---

1) Die „Überlappung“ von Objektmengen wird bei der extensionalen Interpretation von ontologischen Signaturen als Anforderung in Abschnitt 3.1.3.1.1 formuliert.

2) Vgl. ZELEWSKI (1995), Bd. 4, S. 227.

Darüber hinaus können Individuen der Sorte s_{n+1} auf unterschiedliche „Weise“ konstruiert werden. Wenn beispielsweise neben dem o.a. Operationssymbol O_1 mit

$$typ_{OPS_{SS}}(O_1)=(s1_1...s1_n,s_{n+1})$$

ein weiteres Operationssymbol O_2 mit

$$typ_{OPS_{SS}}(O_1)=(s2_1...s2_n,s_{n+1})$$

in der sortierten Signatur SIG_{SS} spezifiziert ist, dann kann sowohl durch die Anwendung der Operation o_1 auf ein Objekttupel $(ob1_1,...,ob1_n)$ mit $(ob1_1,...,ob1_n)\in OB_{s_{1.1}}\times...\times OB_{s_{1.1}}$ als auch durch die Anwendung der Operation o_2 auf ein Objekttupel $(ob2_1,...,ob2_n)$ mit $(ob2_1,...,ob2_n)\in OB_{s_{2.1}}\times...\times OB_{s_{2.1}}$ ein Individuum der Sorte s_{n+1} erzeugt werden.

Die innere Strukturierung von Objekten ist auch schon im Rahmen der konventionellen Prädikatenlogik möglich. Auch dort lassen sich Objekte aus der typgerechten Anwendung von Operationen auf Objekttupel konstruieren. Im Gegensatz zur sortierten Prädikatenlogik kann jedoch im Rahmen der konventionellen Prädikatenlogik nicht näher bestimmt werden, welcher „Art“ die Objekte aus dem Objekttupel sein müssen. Denn die konventionelle Prädikatenlogik geht von einer undifferenzierten Objektmenge OB aus. Insofern kann auch nicht angegeben werden, welcher Art ein Objekttupel sein muss, auf das eine Operation – zwecks strukturierter Konstruktion eines Individuums – angewendet werden kann.

Die Familie OPF der Operationen aus einer SIG_{SS}-Struktur $A_{SIG_{SS}}$ umfasst Operationen, die über den sortenspezifischen Objektmengen definiert sind. Jedes Operationssymbol $O_i\in OPS$ mit $typ_{OPS_{SS}}=(s_1...s_n,s_{n+1})$ aus einer sortierten Signatur SIG_{SS} wird extensional durch eine Operation

$$o_i\in OPF \text{ mit } I_{OPS}(O_i)=o_i$$
$$\text{mit } o_i\text{: } OB_{s_1}\times ... \times OB_{s_n}\rightarrow OB_{s_{n+1}}$$

extensional interpretiert. Die sortenspezifischen Objektmengen $OB_{s_1}\times ... \times OB_{s_n}$ im Argumentbereich von o_i sind genau jenen Sorten $s_1,...,s_n$ zugeordnet, die in der Typisierung $typ_{OPS_{SS}}(O_i)=(s_1,...,s_n,s_{n+1})$ des Operationssymbols O_i angegeben sind.

Jedes null-stellige Operationssymbol $O_i\in OPS$ mit $ARG_{OPS_{SS}}(O_i)=\lambda$ und $ZIEL_{OPS_{SS}}(O_i)=s$ aus einer sortierten Signatur SIG_{SS} wird durch eine wiederum null-stellige Operation $o_i\in OPF$ mit der Operationsvorschrift $o_i{:}\rightarrow OB_s$ aus einer SIG_{SS}-Struktur $A_{SIG_{SS}}$ extensional interpretiert. Während Erstgenannte bereits als Konstantensymbole eingeführt wurden, handelt es sich bei Letztgenannten um *Konstanten*. Konstanten sind Operationen, aus deren Anwendung auf das leere Argument ein Individuum $o_i()=ob_u$ hervorgeht. Das Individuum ist ein formales Objekt aus einer Objektmenge OB_s zu einer Sorte s, die als Zielsorte $ZIEL_{OPS_{OS}}(O_i)$ des Konstantensymbols O_i angegeben ist. Somit können die Extensionen von Konstantensymbolen als Bilder von null-stelligen Operationen bei der Anwendung auf das leere Argument charakterisiert werden.[1)]

1) Vgl. ZELEWSKI (1995), Bd. 4, S. 15.

Die Familie RF aus einer SIG_{SS}-Struktur $A_{SIG_{SS}}$ umfasst alle Relationen, die über den sortenspezifischen Objektmengen definiert sind. Jedes Relationssymbol $R_j \in RS$ mit $typ_{RS_{SS}}=(s_1...s_n)$ aus einer sortierten Signatur SIG_{SS} wird durch eine Relation $r_j \in RF$ mit $r_j \subseteq OB_{s_1} \times ... \times OB_{s_n}$ interpretiert. Wie bei den Operationen aus einer SIG_{SS}-Struktur auch, entspricht der Argumentbereich einer Relation r_j dem kartesischen Produkt sortenspezifischer Objektmengen. Auch hierbei sind es solche Objektmengen, die für diejenigen Sorten spezifisch sind, die in der Typisierung des interpretierten Relationssymbols R_j angegeben sind.

Bei der Zuordnung von Operationen und Relationen zu Operations- bzw. Relationssymbolen wird stets die Typisierung der Letztgenannten durch die Typisierungsfunktionen $typ_{OPS_{SS}}$ bzw. $typ_{RS_{SS}}$ berücksichtigt. Den Komponenten einer sortierten Signatur SIG_{SS} werden durch die Interpretationsfunktionen I_{OPS} und I_{RS} nur solche Komponenten einer SIG_{SS}-Struktur $A_{SIG_{SS}}$ zugeordnet, bei denen die Stelligkeiten und die Sortenstrukturen übereinstimmen. Die Typisierungsfunktionen $typ_{OPS_{SS}}$ und $typ_{RS_{SS}}$ haben daher einen *integritätsbewahrenden* Charakter bei der Interpretation. Mit ihnen werden bereits in der Signatur semantisch motivierte Einschränkungen der Ausdrucksmöglichkeiten formuliert.

Zwar sind auch in der konventionellen Prädikatenlogik integritätsbewahrende metasprachliche Ausdrucksmöglichkeiten vorgesehen, allerdings werden diese im Vergleich zur sortierten Prädikatenlogik weniger restriktiv formuliert. Die integritätsbewahrenden Typisierungsfunktionen $typ_{OPS_{KS}}$ und $typ_{RS_{KS}}$ haben dort nämlich in ihren Zielbereichen die Menge $\mathbb{N}$ aller natürlichen Zahlen. Die Einschränkungen der Interpretationsmöglichkeiten sind daher nur numerischer Art. Bei der Interpretation von Operations- und Relationssymbolen aus konventionellen Signaturen kommen alle Operationen bzw. Relationen in Frage, deren Argumentbereiche die gleiche numerische „Breite“ aufweisen wie das jeweils zu interpretierende Symbol.

Durch den Übergang zu sortierten Signaturen werden *zusätzliche* Integritätsbedingungen bei der extensionalen Interpretation von Operations- und Relationssymbolen formuliert. Bei der extensionalen Interpretation von Operations- und Relationssymbolen aus einer sortierten Signatur SIG_{SS} wird nämlich – neben der numerischen Stelligkeit – auch die Sortenstruktur, d.h. die *Sortengerechtheit* der Objektmengen in den Argumentbereichen der Operationen bzw. Relationen überprüft. Den Typisierungsfunktionen $typ_{OPS_{SS}}$ und $typ_{RS_{SS}}$ kommt dadurch ein stärkerer Restriktionscharakter zu als den Typisierungsfunktionen $typ_{OPS_{KS}}$ und $typ_{RS_{KS}}$ aus konventionellen Signaturen.

Die Menge $A(SIG_{SS})$ umfasst alle Strukturen, mit denen eine sortierte Signatur SIG_{SS} extensional interpretiert werden kann. Jedes Element $A_{SIG_{SS}} \in A(SIG_{SS})$ beinhaltet demnach eine Familie OBF_{SS} sortenspezifischer Objektmengen, eine Familie OPF von Operationen, eine Familie RF von Relationen und eine Familie IF_{SS} von Interpretationsfunktionen. Dabei ist jedes Element der Menge $A(SIG_{SS})$ „passend“ zu der sortierten Signatur SIG_{SS} aufgebaut. Das heißt, dass es sich bei den Interpretationsfunktionen I_S, I_{OPS} und I_{RS} aus jeder SIG_{SS}-Struktur $A_{SIG_{SS}} \in A(SIG_{SS})$ um linkstotale, bijektive Funktionen handelt.

2.2.1.2.3.2 Auswertung von sortierten Ausdrücken

2.2.1.2.3.2.1 Auswertung von sortierten Termen

Die Verknüpfung einer sortierten Signatur SIG_{SS} mit einer SIG_{SS}-Struktur $A_{SIG_{SS}}$ erfolgt einerseits durch die Interpretationsfunktionen I_S, I_{OPS} und I_{RS}. Durch sie werden den Komponenten einer sortierten Signatur SIG_{SS} jeweils formale Konstrukte aus einer SIG_{SS}-Struktur $A_{SIG_{SS}}$ zugeordnet. Um auch sortierte Ausdrücke über einer sortierten Signatur SIG_{SS} auswerten zu können, bedarf es zum einen eines Verfahrens zur Auswertung von sortierten Termen. Zum anderen bedarf es eines Verfahrens zur Auswertung von sortierten Formeln.

Die Auswertung sortierter Terme erfolgt durch die Mitglieder einer Familie

$$ITF_{SS}=(IT_s)_{s\in S}$$

sortenspezifischer Termauswertungsfunktionen. Dabei ist jede sortenspezifische Termauswertungsfunktion IT_s definiert als:

$$IT_s\text{: } TERM_s \rightarrow OB_s.$$

Jede Termauswertungsfunktion IT_s ordnet jedem Term $t\in TERM_s$ zu einer Sorte $s\in S$ genau ein Individuum $IT_s(t)=ob_u$ aus dem Objektbereich OB_s zur selben Sorte s zu.

Während die Auswertung von konventionellen Termen unsortiert verläuft, wird bei sortierten Termen zunächst ihre Sorte überprüft. Je nachdem, welcher sortenspezifischen Termmenge $TERM_s\in TERMF_{SIG_{SS}}$ ein sortierter Term t zugehört, wird eine sortenspezifische Termauswertungsfunktion IT_s angewendet. Der Bildbereich jeder sortenspezifischen Termauswertungsfunktion IT_s liegt stets in der sortenspezifischen Objektmenge OB_s, die zu derselben Sorte s gehört, wie die Termmenge $TERM_s$, deren Elemente ausgewertet werden. Verdeutlicht wird dieser Zusammenhang durch den Sortenindex „s“, der jeder sortenspezifischen Termauswertungsfunktion IT_s beigefügt ist. Der Sortenindex „s“ einer Auswertungsfunktion IT_s schränkt die Menge der Terme im Argumentbereich der Funktionsvorschrift auf s-spezifische Terme ein. Durch die Identität der Indizes für die Menge $TERM_s$ der Terme im Vor- und die Menge OB_s der formalen Objekte im Nachbereich der Termauswertungsfunktion IT_s ist gewährleistet, dass die Auswertung $IT_s(t)$ eines Terms $t\in TERM_s$ zu einem Individuum ob_u führt, das Element der Objektmenge OB_s zur selben Sorte s ist.[1)]

Für die Auswertung variabler Terme ist zusätzlich die *Belegung* der Variablen mit Individuen aus sortenspezifischen Objektmengen einer SIG_{SS}-Struktur $A_{SIG_{SS}}$ notwendig. Die Variablenbelegung erfolgt durch die Mitglieder $bel_{s_1},...,bel_{s_n}$ einer Familie $belf_{SS}$ sortenspezifischer Variablenbelegungsfunktionen:

$$belf_{SS} = (bel_s\text{:}VAR_s \rightarrow OB_s)_{s\in S}.$$

1) Dieser Aspekt der Termauswertung wird als ihre „Sortentreue“ bezeichnet; vgl. ZELEWSKI (1995), Bd. 4, S. 45.

Bei der Belegung einer Variable wird jeder sortenspezifischen Variable $x_q \in VAR_s$ ein formales Objekt $ob_u \in OB_s$ zur selben Sorte s zugeordnet. Analog zu den sortenspezifischen Termauswertungsfunktionen IT_s sind die sortenspezifischen Variablenbelegungen bel_s für jede sortenspezifische Variablenmenge VAR_s definiert. Der Bildbereich jeder Variablenbelegung bel_s ist die sortenspezifische Objektmenge OB_s, die zu derselben Sorte s gehört, wie die Variablenmenge VAR_s, deren Elemente belegt werden. Die *Sortentreue* der Funktionsanwendung wird – wie bei den Termauswertungsfunktionen auch – durch den Sortenindex „s“ der Variablenbelegung gewährleistet. Dadurch können Variablen nur mit formalen Objekten aus solchen Objektmengen belegt werden, deren Sortenzugehörigkeit sich mit der Sortenzugehörigkeit der Variablenmenge deckt, aus der die Variable stammt.

Um nachher quantifizierte Formeln auswerten zu können, wird auch bei Variablen über sortierten Signaturen eine bedingte Variablenbelegung benötigt. Jede *bedingte sortenspezifische Variablenbelegungsfunktion*

$$bel_s[x^*/ob_u](x) = \begin{cases} bel_s(x) & \text{wenn } x \neq x^* \\ ob_u & \text{wenn } x = x^* \end{cases}$$

weist jeder Variablen x aus der sortenspezifischen Variablenmenge VAR_s ein formales Objekt aus der sortenspezifischen Objektmenge OB_s entsprechend der Variablenbelegung bel_s zu, wenn sich die Variable x von einer näher spezifizierten Variable x* unterscheidet.

Die Mitglieder der Familie ITF_{SS} aller sortenspezifischen Termauswertungsfunktionen sind bei gegebener Variablenbelegung bel_s für alle Sorten $s \in S$ und einer vorgegebenen SIG_{SS}-Struktur $A_{SIG_{SS}}$ rekursiv wie folgt definiert:

(1.) $IT_s(x_q)=bel_s(x_q)=ob_u$ mit $ob_u \in OB_s$
für jede Variable $x \in VAR_s$ und jede Sorte $s \in S$,

(2.) $IT_s(O_i)=I_{OPS}(O_i)=o_i$ mit $o_i()=ob_u$ und $ob_u \in OB_s$
für jedes Konstantensymbol $O_i \in OPS$ mit $typ_{OPS_{SS}}(O_i)=(\lambda,s)$ und

(3.) $IT_{s_{n+1}}(O_i(t_1...t_n))=o_i(IT_{s_1}(t_1),...,IT_{s_n}(t_n))$
für jedes Operationssymbol $O_i \in OPS$ mit $typ_{OPS_{SS}}(O_i)=(s_1...s_n,s_{n+1})$ mit den Sorten $k_1,...,k_n,k_{n+1} \in K$ und den Termen $t_1,...,t_n$ mit $t_x \in TERM_{k_x}$ für $x=1,...,n$ und $n \in \mathbb{N}_+$.

Die Rekursionsbasis für die sortenspezifische Termauswertung wird durch die ersten beiden Regeln der Termauswertung gelegt.

Mit der ersten Regel wird festgelegt, dass die Auswertung eines sortierten Terms $t \in TERM_{SIG_{SS}}$, der aus einer sortierten Variable x hervorgeht, zu dem formalen Objekt ob_u führt, mit dem x durch eine sortenspezifische Variablenbelegung bel_s belegt ist.

Durch die zweite Regel wird festgelegt, dass die Auswertung eines Konstantensymbols $O_i \in OPS$ mit $typ_{OPS_{SS}}(O_i)=(\lambda,s)$ zu der Konstanten $I_{OPS}(O_i)=o_i$ mit $o_i()=ob_u$ und $ob_u \in OB_s$ führt. Die Auswertung null-stelliger Operationssymbole mittels IT_s deckt sich somit mit ihrer extensionalen Interpretation mittels der Interpretationsfunktion I_{OPS}. Bei jedem

Konstantensymbol handelt es sich nämlich um ein null-stelliges Operationssymbol. Wie alle anderen Operationssymbole aus einer sortierten Signatur SIG_{SS} auch, werden null-stellige Operationssymbole durch die Interpretationsfunktion I_{OPS} auf jeweils eine Operation abgebildet. Die Bilder der Funktion I_{OPS} zu Konstantensymbolen sind stets Konstanten. Bei Konstanten handelt es sich wiederum um Operationen, deren Anwendung auf das leere Argument ein Individuum $o_i()=ob_u$ hervorbringt.

Variablen und Konstantensymbole werden mittels einer sortenspezifischen Termauswertungsfunktion IT_s jeweils auf *atomare* Individuen abgebildet. Dies wird durch die ersten beiden Regeln, die zur Auswertung von Termen definiert sind, gewährleistet. Zusammengesetzte Terme, die aus der typgerechten Anwendung von Operationssymbolen auf n-Tupel von Termen hervorgehen, werden hingegen auf *zusammengesetzte* Individuen, die aus der Anwendung von Operationen auf n-Tupel von formalen Objekten hervorgehen, abgebildet. Dadurch haben Terme einerseits und die Individuen, zu denen sie ausgewertet werden, andererseits stets die gleiche *Struktur*. Entweder handelt es sich um atomare Individuen, zu denen atomare Terme ausgewertet werden können, oder es handelt sich um zusammengesetzte Individuen, zu denen zusammengesetzte Terme ausgewertet werden können. Bei der zweiten Alternative haben zusammengesetzte Terme stets die gleiche Struktur wie die zusammengesetzten Individuen, zu denen sie ausgewertet werden können.

2.2.1.2.3.2.2 Auswertung von sortierten Formeln

Die zweite Teilmenge der Menge $EXPR_{SIG_{SS}}$ aller Ausdrücke über einer sortierten Signatur SIG_{SS} beinhaltet sortierte Formeln. Für die Auswertung sortierter Formeln wird – wie bei der konventionellen Prädikatenlogik auch – die Modellrelation $\models$ verwendet. Genau dann, wenn eine sortierte Formel $F \in FORM_{SIG_{SS}}$ durch die Mitglieder einer Familie $belf_{SS}$ sortenspezifischer Variablenbelegungen in einer SIG_{SS}-Struktur $A_{SIG_{SS}}$ bestätigt wird, gilt:

$$(A_{SIG_{SS}}, belf_{SS}) \models F$$

Dabei ist die Modellrelation $\models$ für Formeln über sortierten Signaturen wie folgt definiert:

(1.) $(A_{SIG_{SS}}, belf_{SS}) \models w$ gilt für *jedes* Tupel $(A_{SIG_{SS}}, belf_{SS})$,

(2.) $(A_{SIG_{SS}}, belf_{SS}) \models f$ gilt für *kein* Tupel $(A_{SIG_{SS}}, belf_{SS})$,

(3.) $(A_{SIG_{SS}}, belf_{SS}) \models R_j(t_1,...,t_n) \Leftrightarrow (IT_{s_1}(t_1),...,IT_{s_n}(t_n)) \in r_j$ mit $I_{RS}(R_j)=r_j$,

(4.) $(A_{SIG_{SS}}, belf_{SS}) \models \neg F \Leftrightarrow (A_{SIG_{SS}}, belf_{SS}) \not\models F)$,

(5.) $(A_{SIG_{SS}}, belf_{SS}) \models F_1 \wedge F_2 \Leftrightarrow (A_{SIG_{SS}}, belf_{SS}) \models F_1$ und $(A_{SIG_{SS}}, belf_{SS}) \models F_2$,

(6.) $(A_{SIG_{SS}}, belf_{SS}) \models F_1 \vee F_2 \Leftrightarrow (A_{SIG_{SS}}, belf_{SS}) \models F_1$ oder $(A_{SIG_{SS}}, belf_{SS}) \models F_2$, (7.) $(A_{SIG_{SS}},belf_{SS}) \vDash F_1 \underline{\vee} F_2$ $\Leftrightarrow$ entweder $(A_{SIG_{SS}},belf_{SS}) \vDash F_1$ oder $(A_{SIG_{SS}},belf_{SS}) \vDash F_2$,

(8.) $(A_{SIG_{SS}},belf_{SS}) \vDash F_1 \rightarrow F_2$ $\Leftrightarrow$ $(A_{SIG_{SS}},belf_{SS}) \nvDash F_1)$ oder $(A_{SIG_{SS}},belf_{SS}) \vDash F_2$

(9.) $(A_{SIG_{SS}},belf_{SS}) \vDash F_1 \leftrightarrow F_2$ $\Leftrightarrow$ $(A_{SIG_{SS}},belf_{SS}) \vDash F_1 \rightarrow F_2$ und $(A_{SIG_{SS}},belf_{SS}) \vDash F_2 \rightarrow F_1$

(10.) $(A_{SIG_{SS}},belf_{SS}) \vDash \forall x\text{: } F$ $\Leftrightarrow$ $(A_{SIG_{SS}},belf_{SS}[x/ob_u]) \vDash F$ für alle $ob_u \in OB_s$

(11.) $(A_{SIG_{SS}},belf_{SS}) \vDash \exists x\text{: } F$ $\Leftrightarrow$ $(A_{SIG_{SS}},belf_{SS}[x/ob_u]) \vDash F$ für mindestens ein $ob_u \in OB_s$

(12.) $(A_{SIG_{SS}},belf_{SS}) \vDash \underline{\exists} x\text{: } F$ $\Leftrightarrow$ $(A_{SIG_{SS}},belf_{SS}[x/ob_u]) \vDash F$ für genau ein $ob_u \in OB_s$

Wenn eine sortierte Formel $F \in FORM_{SIG_{SS}}$ durch jede Familie $belf_{SS}$ sortenspezifischer Variablenbelegungen in einer SIG_{SS}-Struktur $A_{SIG_{SS}}$ bestätigt wird, dann wird F als *gültig* in der SIG_{SS}-Struktur $A_{SIG_{SS}}$ bezeichnet. Die SIG_{SS}-Struktur $A_{SIG_{SS}}$ ist in diesem Fall ein *Modell* der sortierten Formel F. Um die Gültigkeit einer Formel $F \in FORM_{SIG_{SS}}$ in einer SIG_{SS}-Struktur $A_{SIG_{SS}}$ auszudrücken, wird die Schreibweise

$$A_{SIG_{SS}} \vDash F$$

verwendet.

Eine Menge $FM \subseteq FORM_{SIG_{SS}}$ sortierter Formeln mit $FM=\{F_1,...,F_x\}$ wird durch eine Familie $belf_{SS}$ sortenspezifischer Variablenbelegungen in einer SIG_{SS}-Struktur $A_{SIG_{SS}}$ bestätigt, wenn die Formel $F_1 \wedge ... \wedge F_x$ durch $belf_{SS}$ in $A_{SIG_{SS}}$ bestätigt wird. Somit gilt:

$$(A_{SIG_{SS}},belf_{SS}) \vDash FM$$
mit $FM \subseteq FORM_{SIG_{SS}}$ und $FM=\{F_1,...,F_x\}$ genau dann, wenn gilt:
$$(A_{SIG_{KS}},belf_{SS}) \vDash F_1 \wedge ... \wedge F_x.$$

Eine Menge FM sortierter Formeln wird als gültig in einer SIG_{SS}-Struktur $A_{SIG_{SS}}$ bezeichnet, wenn die konjunktive Verknüpfung $F_1 \wedge ... \wedge F_x$ aller Formeln $F_1,...,F_x \in FM$ in $A_{SIG_{SS}}$ gültig ist. Somit gilt:

$$A_{SIG_{SS}} \vDash FM$$
mit $FM \subseteq FORM_{SIG_{SS}}$ und $FM=\{F_1,...,F_x\}$ genau dann, wenn gilt:
$$A_{SIG_{SS}} \vDash F_1 \wedge ... \wedge F_x.$$

Für die Modellrelation $\vDash$ gelten die Ausführungen, die bereits in Abschnitt 2.2.1.1.3.2.2 im Rahmen der konventionellen Prädikatenlogik vorgestellt wurden. Analog dazu kann mit der Funktion MOD_{SS} die Menge aller Modelle zu einer sortierten Formel $F \in FORM_{SIG_{SS}}$ bestimmt werden:

$$MOD_{SS}\text{: } FORM_{SIG_{SS}} \rightarrow pot(A(SIG_{SS}))$$
$$\text{mit } MOD_{SS}(F) = \{A_{SIG_{SS}} \mid (A_{SIG_{SS}} \vDash F)\}$$

Ebenso gelten die Ausführungen aus Abschnitt 2.2.1.1.3.2.2 zu Formelmengen analog. So gilt für eine Formelmenge $FM \subseteq FORM_{SIG_{SS}}$ mit $FM=\{F_1,...,F_x\}$:

$$MOD_{SS}(FM) = \{A_{SIG_{SS}} \mid (A_{SIG_{SS}} \vDash (F_1 \wedge ... \wedge F_x))\}$$
$$\text{und somit } MOD_{SS}(FM) = MOD_{SS}(F_1 \wedge ... \wedge F_x).$$

Demnach umfasst die Menge $MOD_{SS}(FM)$ alle SIG_{SS}-Strukturen, die Modelle der Formel $F_1 \wedge ... \wedge F_x$ sind, die aus der konjunktiven Verknüpfung aller Formeln aus FM hervorgeht.

Eine sortierte Formel $F \in FORM_{SIG_{SS}}$ ist genau dann *erfüllbar*, wenn es mindestens eine SIG_{SS}-Struktur $A_{SIG_{SS}}$ gibt, in der sie durch mindestens eine Familie $belf_{SS}$ sortenspezifischer Variablenbelegungen bestätigt wird. Entsprechend ist eine Formelmenge $FM \subseteq FORM_{SIG_{SS}}$ genau dann erfüllbar, wenn es mindestens eine SIG_{SS}-Struktur $A_{SIG_{SS}}$ gibt, in der die Konjunktion $F_1 \wedge ... \wedge F_X$ aller sortierten Formeln $F_1,...,F_x \in FM$ von mindestens einer Familie $belf_{SS}$ sortenspezifischer Variablenbelegungen bestätigt werden kann.[1)]

Eine sortierte Formel $F \in FORM_{SIG_{KS}}$ ist *allgemeingültig* oder *tautologisch*, wenn sie von *jeder* Familie $belf_{SS}$ sortenspezifischer Variablenbelegungen in *jeder* SIG_{KS}-Struktur $A_{SIG_{KS}} \in A(SIG_{KS})$ bestätigt wird. Entsprechend gilt für allgemeingültige sortierte Formeln, dass sie in jeder SIG_{SS}-Struktur $A_{SIG_{SS}}$ gültig sind. Da somit bei einer allgemeingültigen sortierten Formeln F jede SIG_{SS}-Struktur $A_{SIG_{SS}}$ ein Modell von F ist, gilt in diesem Fall

$$A(SIG_{SS}) = MOD_{SS}(F).$$

Demnach sind bei einer allgemeingültigen sortierten Formel $F \in FORM_{SIG_{SS}}$ die Menge $A_{SIG_{SS}}(SIG_{SS})$ aller SIG_{SS}-Strukturen und die Menge $MOD_{SS}(F)$ aller Modelle von F gleich.

Eine sortierte Formel $F \in FORM_{SIG_{KS}}$ ist genau dann *widersprüchlich*, wenn keine SIG_{SS}-Struktur $A_{SIG_{SS}}$ und keine Familie $belf_{SS}$ sortenspezifischer Variablenbelegungen existieren, so dass die F durch $belf_{SS}$ in $A_{SIG_{SS}}$ bestätigt wird. Widersprüchliche sortierte Formeln zeichnen sich dadurch aus, dass es keine SIG_{SS}-Struktur $A_{SIG_{SS}}$ gibt, die sie als Modell haben. Eine Menge $FM=\{F_1,...,F_x\}$ sortierter Formeln ist genau dann widersprüchlich, wenn die konjunktive Verknüpfung $F_1 \wedge ... \wedge F_x$ aller Formeln aus FM widersprüchlich ist.

Um auszudrücken, dass die Menge $MOD_{SS}(F_1)$ aller Modelle zu einer sortierten Formel $F_1 \in FORM_{SIG_{SS}}$ eine Teilmenge der Menge $MOD_{SS}(F_2)$ aller Modelle zu einer sortierten Formel $F_2 \in FORM_{SIG_{SS}}$ ist, wird auch im Rahmen der sortierten Prädikatenlogik die

1) Analog zu der der Menge $CF_{SIG_{KS}}$ aller geschlossenen Formeln über einer konventionellen Signatur SIG_{KS} ist bei den Elementen der Menge $CF_{SIG_{SS}}$ aller geschlossener Formeln über einer sortierten Signatur SIG_{SS} ihre Erfüllbarkeit von der Variablenbelegung unabhängig. Für die Erfüllbarkeit geschlossener sortierter Formeln ist lediglich die SIG_{SS}-Struktur $A_{SIG_{SS}}$, die zur extensionalen Interpretation der sortierten Signatur SIG_{SS} herangezogen wird, von Relevanz.

Folgerungsrelation $\Vdash$ verwendet. Im Gegensatz zur Modellrelation $\vDash$, die dazu verwendet wird, die Bestätigung oder die Gültigkeit einer sortierten Formel in einer SIG_{SS}-Struktur $A_{SIG_{SS}}$ auszudrücken, weist die Folgerungsrelation $\Vdash$ keinen unmittelbaren Bezug zu SIG_{SS}-Strukturen auf. Sie verbindet jeweils die Gültigkeit einer Formel(-menge) mit der Gültigkeit einer weiteren Formel(-menge).

Die Folgerungsbeziehung zwischen zwei sortierten Formeln $F_1,F_2 \in FORM_{SIG_{SS}}$ ist genau dann erfüllt, wenn die Menge $MOD_{SS}(F_1)$ aller Modelle der sortierten Formel F_1 eine Teilmenge der Menge $MOD_{SS}(F_2)$ aller Modelle der sortierten Formel F_2 ist:[1)]

$$\forall F_1, F_2 \in FORM_{KS}: MOD_{KS}(F_1) \subseteq MOD_{KS}(F_2) \leftrightarrow F_1 \Vdash F_2.$$

Die Folgerungsrelation $\Vdash$ wird – wie die Modellrelation $\vDash$ auch – dadurch „überladen“, dass sie auch für Formelmengen verwendet wird:

$$\forall FM_1, FM_2 \subseteq FORM_{KS}: MOD_{KS}(FM_1) \subseteq MOD_{KS}(FM_2) \leftrightarrow FM_1 \Vdash FM_2.$$

Eine Formelmenge FM_2 folgt demnach genau dann aus einer Formelmenge FM_1, wenn die Menge $MOD_{SS}(FM_1)$ aller Modelle von FM_1 eine Teilmenge der Menge $MOD_{SS}(FM_2)$ aller Modelle von FM_2 ist.

Für die Eigenschaften der Folgerungsrelation $\Vdash$ gelten ansonsten die Eigenschaften analog, die in 2.2.1.1.3.2.2 definiert wurden. Hierzu zählt insbesondere die Reflexivität, Transitivität und Monotonie der Folgerungsrelation $\Vdash$. Darüber hinaus wird für den Spezialfall einer symmetrischen Folgerungsbeziehung zwischen zwei Formelmengen das Symbol $\equiv$ für die logische *Äquivalenz* sortierter Formeln verwendet. Dabei ist die logische Äquivalenz zwischen zwei sortierten Formeln $F_1,F_2 \in FORM_{SIG_{SS}}$ genau dann erfüllt, wenn sie wechselseitig in Folgerungsrelation $\Vdash$ zueinander stehen. In diesem Fall sind die Mengen $MOD_{SS}(F_1)$ und $MOD_{SS}(F_2)$ der Modellmenge von F_1 bzw. F_2 gleich:

$$\forall F_1,F_2 \in FORM_{SIG_{SS}}: ((F_1 \Vdash F_2) \wedge (F_2 \Vdash F_1)) \leftrightarrow (F_1 \equiv F_2).$$

1) Vgl. EBBINGHAUS ET AL. (1992), S. 41; EHRIG ET AL. (1999), S. 337 f.

2.2.1.3 Prädikatenlogische Spezifikationen

Konventionelle und sortierte Signaturen werden dazu verwendet, die deskriptiven Symbole aus dem jeweils zugrunde liegenden formalsprachlichen Alphabet $ALPH_{KS}$ bzw. $ALPH_{SS}$ derart zu typisieren, dass korrekte prädikatenlogische Ausdrücke konstruiert werden können. Eine prädikatenlogische Signatur[1)] SIG_{KS} oder SIG_{SS} gibt einen formalen Rahmen vor, dem eine Struktur $A_{SIG_{KS}} \in A(SIG_{KS})$ bzw. $A_{SIG_{SS}} \in A(SIG_{SS})$ entsprechen muss, um die jeweilige Signatur extensional interpretieren zu können.

Über die Deklaration und Typisierung der deskriptiven Symbole hinaus werden in prädikatenlogischen Signaturen allerdings keine inhaltlichen Einschränkungen für Strukturen getroffen, mit denen sie extensional interpretiert werden können. Die Interpretation einer prädikatenlogischen Signatur SIG_{PL}[2)] durch eine SIG_{PL}-Struktur $A_{SIG_{PL}}$ entspricht daher der „*losen*" Semantik von SIG_{PL}, wenn keine weiteren Einschränkungen definiert sind.[3)] Sie ist in dem Sinne lose, als dass in $A(SIG_{PL})$ auch solche SIG_{PL}-Strukturen enthalten sein können, die keine zulässigen Interpretationen der zugrunde liegenden Signatur SIG_{PL} darstellen. Beispielsweise kann es im Fall einer SIG_{KS}-Struktur $A_{SIG_{KS}}$ zu einer konventionellen Signatur SIG_{KS} sein, dass die Relationssymbole

$$\text{Unternehmen, Person} \in RS$$

nicht miteinander disjunkte Extensionen aufweisen. Analog kann eine sortierte Signatur SIG_{SS}, in der die Sorten *Unternehmen* und *Person* verwendet werden, durch eine solche SIG_{SS}-Struktur $A_{SIG_{SS}}$ extensional interpretiert werden, in der die sortenspezifischen Objektmengen $OB_{Unternehmen}$ und OB_{Person} nicht miteinander disjunkt sind.

Um eine weitere Eingrenzung der in Frage kommenden Strukturen zu erreichen, ist es notwendig, eingrenzende Ausdrücke zu formulieren, die den Charakter von *Anforderungen* an die entsprechenden SIG_{PL}-Strukturen haben. Mit den Mengen $FORM_{SIG_{KS}}$ und $FORM_{SIG_{SS}}$ wurden in den vorherigen Abschnitten Formeln über konventionellen bzw. sortierten Signaturen vorgestellt, die hierzu verwendet werden können. Zwecks Eingrenzung der entsprechenden Menge aller in Frage kommenden SIG_{PL}-Strukturen zu einer prädikatenlogischen Signatur SIG_{PL} wird eine Teilmenge der Menge $FORM_{SIG_{PL}}$ aller prädikatenlogischen Formeln herangezogen, die als „Regeln" bezeichnet werden.[4)]

---

1) Im Folgenden werden die Bezeichnungen „prädikatenlogische Signatur" und „prädikatenlogische Spezifikation" verwendet, wenn es für die Argumentation nicht von Bedeutung ist, ob es sich um eine entweder konventionelle oder sortierte Signatur bzw. Spezifikation handelt.

2) Mit dem Index PL wird verdeutlicht, dass es sich um eine prädikatenlogische Signatur handelt, bei der zunächst unbestimmt bleibt, ob es sich um eine konventionelle oder sortierte Signatur handelt. Formal kann der Ausdruck PL als Variable mit $PL \in \{KS,SS\}$ aufgefasst werden.

3) Vgl. EHRIG ET AL. (1999), S. 182.

4) Eine weitere Unterteilung des Begriffs „Regeln" erfolgt im Kontext ontologischer Spezifikationen in Abschnitt 3.1.4.1. Dort werden *Inferenz-* und *Integritätsregeln* mit ihrem jeweiligen zweckorientierten Hintergrund vorgestellt.

Eine prädikatenlogische Signatur SIG_{PL} in Verbindung mit einer Menge $REG_{SIG_{PL}}$ von Regeln, die mit den Ausdrucksmitteln aus der prädikatenlogischen Signatur SIG_{PL} konstruiert wurden, wird als *prädikatenlogische Spezifikation* bezeichnet.

Prädikatenlogische Spezifikationen sind zwar sowohl für konventionelle als auch für sortierte Signaturen denkbar. Allerdings werden sie in der Regel lediglich mit Zweitgenannten assoziiert.[1] Diese Eingrenzung des Prinzips prädikatenlogischer Spezifikationen ist allerdings nicht denknotwendig. Daher werden im Folgenden unter der Bezeichnung *prädikatenlogische Spezifikationen* Erweiterungen von prädikatenlogischen Signaturen vorgestellt. Dabei lässt es die Bezeichnung *prädikatenlogische Spezifikation* offen, ob eine konventionelle oder sortierte Signatur zugrunde gelegt wird.

Die Erweiterung einer prädikatenlogischen Signatur SIG_{PL} um eine Menge $REG_{SIG_{PL}}$ von Regeln wird als *prädikatenlogische*[2] *Spezifikation* $SPEZ_{PL}$ bezeichnet.[3] Sie ist definiert als:

$$SPEZ_{PL} = (SIG_{PL}, REG_{SIG_{PL}})$$

Die Komponenten einer prädikatenlogischen Spezifikation $SPEZ_{PL}$ sind:

(1.) eine *prädikatenlogische Signatur* SIG_{PL} und

(2.) eine Menge von *Regeln* $REG_{SIG_{PL}}$ mit $REG_{SIG_{PL}} \subseteq FORM_{SIG_{PL}}$.

Für eine prädikatenlogische Spezifikation $SPEZ_{PL}$ wird eine prädikatenlogische Signatur SIG_{PL} vorausgesetzt. Es kann sich dabei entweder um eine konventionelle Signatur (SIG_{KS}) oder um eine sortierte Signatur (SIG_{SS}) handeln. Je nachdem, welche Signatur-Art als erste Komponente bestimmt ist, handelt es sich bei den Elementen der zweiten Komponente um konventionelle oder sortierte Formeln. Wenn es sich bei der ersten Komponente einer prädikatenlogischen Spezifikation $SPEZ_{PL}$ um eine konventionelle Signatur SIG_{KS} handelt, sind die Elemente von $REG_{SIG_{PL}}$ konventionelle Formeln. Sie werden entsprechend der induktiven Grammatik zur Konstruktion konventioneller For-

1) Vgl. KREOWSKI (1991), S. 65.

2) In der Regel wird der Zusatz „prädikatenlogische“ ausgelassen; vgl. z.B. EHRIG ET AL. (1999), S. 11. Eine Unterscheidung wird lediglich hinsichtlich des Vorhandenseins von Relationssymbolen in der sortierten Signatur zwischen *algebraischen* und *relationale* Spezifikationen vorgenommen. Algebraische Spezifikationen gehen aus der Erweiterung algebraischer Signaturen um eine Menge von Regeln hervor. Da in algebraischen Signaturen keine Relationssymbole vorgesehen sind, ist es im Grunde auch nicht möglich, über algebraischen Signaturen Formeln zu konstruieren. Als einzige Ausdrucksart kommen lediglich Terme in Frage. In der Regel wird diesem Umstand dadurch begegnet, dass das Relationssymbol „=“ oder ein entsprechendes Äquivalent (z.B. „equal“) zugelassen wird, mit dem sich Gleichungen konstruieren lassen; vgl. EHRICH ET AL. (1989), S. 19 f. Relationale Spezifikationen stimmen mit prädikatenlogischen Spezifikationen überein.
Der Zusatz „prädikatenlogische“ ist hier notwendig, um eine Abgrenzung zu *ontologischen Spezifikationen* vornehmen zu können, die später vorgestellt werden; vgl. Abschnitt 3.1.4.1.

3) Vgl. KREOWSKI (1991), S. 67.

meln konstruiert.[1)] Wenn es sich bei der ersten Komponente hingegen um eine sortierte Signatur SIG_{SS} handelt, sind die Elemente von $REG_{SIG_{PL}}$ sortierte Formeln. Sie werden entsprechend der Grammatik zur Konstruktion sortierter Formeln konstruiert.[2)]

Die Menge $REG_{SIG_{PL}}$ aller Regeln aus einer prädikatenlogischen Spezifikation $SPEZ_{PL}$ mit zugrunde liegender prädikatenlogischer Signatur SIG_{PL} kann grundsätzlich auch leer sein. In diesem Fall degeneriert die prädikatenlogische Spezifikation $SPEZ_{PL}$ zu der prädikatenlogischen Signatur SIG_{PL}. Zwecks begrifflicher Abgrenzung wird vereinbart, prädikatenlogische Spezifikationen mit einer nicht-leeren Menge $REG_{SIG_{PL}}$ als *prädikatenlogische Spezifikationen i.w.S.* zu bezeichnen. Von dem Begriff *prädikatenlogische Spezifikationen i.e.S.* werden auch alle prädikatenlogischen Spezifikationen umfasst, die eine leere Menge $REG_{SIG_{PL}}$ aufweisen und somit mit prädikatenlogischen Signaturen übereinstimmen. Wenn im weiteren Verlauf die Zusätze „i.e.S." oder „i.w.S." ausgelassen werden, ist stets eine prädikatenlogische Spezifikation i.w.S. gemeint.

Eine prädikatenlogische Spezifikation $SPEZ_{PL}$ dient dazu, die Menge $A(SIG_{PL})$ aller Strukturen zu der prädikatenlogischen Signatur SIG_{PL}, die in $SPEZ_{PL}$ enthalten ist, in zwei zueinander disjunkte Teilmengen zu unterteilen. Während die eine Teilmenge aus Modellen zu der Formelmenge $REG_{SIG_{PL}}$ besteht, sind Elemente aus der dazu komplementären Teilmenge von $A(SIG_{PL})$ keine Modelle der Formelmenge $REG_{SIG_{PL}}$. Eine SIG_{PL}-Struktur $A_{SIG_{PL}} \in A(SIG_{PL})$ wird daher auch als Modell der prädikatenlogischen Spezifikation $SPEZ_{PL}=(SIG_{PL},REG_{SIG_{PL}})$ bezeichnet, wenn die Formelmenge $REG_{SIG_{PL}}$ in $A_{SIG_{PL}}$ gültig ist.[3)] Damit eine SIG_{PL}-Struktur $A_{SIG_{PL}}$ ein Modell einer prädikatenlogischen Spezifikation $SPEZ_{PL}=(SIG_{PL},REG_{SIG_{PL}})$ sein kann, muss somit gelten:

$$A_{SIG_{PL}} \models REG_{SIG_{PL}}.$$

Die Modellrelation $\models$ besteht zwischen einer SIG_{PL}-Struktur $A_{SIG_{PL}}$ und der Formelmenge $REG_{SIG_{PL}}$ – entsprechend der Vereinbarung im vorherigen Abschnitt – genau dann, wenn alle Formeln $F \in REG_{SIG_{PL}}$ in $A_{SIG_{PL}}$ gültig sind. Die Gültigkeit einer Formel F in einer SIG_{PL}-Struktur $A_{SIG_{PL}}$ ist wiederum genau dann gegeben, wenn F von jeder Variablenbelegung bel_{KS} bzw. von jeder Variablenbelegungsfunktion aus der Familie $belf_{SS}$ sortenspezifischer Variablenbelegungen in $A_{SIG_{PL}}$ bestätigt wird. Bei Gültigkeit einer Menge FM prädikatenlogischer Formeln in einer SIG_{PL}-Struktur $A_{SIG_{PL}}$ wird letztgenannte als Modell der Formelmenge FM bezeichnet. Alle Modelle zu einer prädikatenlogischen Formelmenge $FM \subseteq FORM_{SIG_{PL}}$ werden wiederum von der Menge $MOD_{PL}(FM)$ umfasst. Die Menge der Modelle zu einer prädikatenlogischen Spezifikation stimmt somit mit der Menge $MOD_{PL}(REG_{SIG_{PL}})$ aller Modelle der Regelmenge $REG_{SIG_{PL}}$ überein. Die Elemente der Formelmenge $REG_{SIG_{PL}}$ haben somit den Charakter

1) Vgl. Abschnitt 2.2.1.1.2.2.2.

2) Vgl. Abschnitt 2.2.1.2.2.2.2.

3) Vgl. EHRICH ET AL. (1989), S. 22; EHRIG ET AL. (1999), S. 186; KREOWSKI (1991), S. 72; LOECKX ET AL. (1996), S. 86.

von Restriktionen, die alle in einer SIG_{PL}-Struktur $A_{SIG_{PL}} \in A(SIG_{PL})$ erfüllt sein müssen, damit $A_{SIG_{PL}}$ ein Modell der prädikatenlogischen Spezifikation $SPEZ_{PL}$ sein kein.

Eine prädikatenlogische Spezifikation $SPEZ_{PL}$ kann schematisch aufgeführt werden, indem für jede ihrer Subkomponenten eine eigene Sektion vorgesehen wird. In der Sektion „S“ werden beispielsweise – im Fall einer sortierten Spezifikation $SPEZ_{SS}$ – alle Sorten aus der sortierten Signatur SIG_{SS} aufgeführt. Für jede weitere Komponente wird eine eigene Sektion aufgeführt. In der folgenden Darstellung ist schematisch eine sortierte Spezifikation wiedergegeben:

$\underline{\underline{\mathrm{Spez}}}_{SS}$

$\underline{\underline{\mathrm{SIG}}}_{SS}$

OPS:

$O_1: \quad s_{O1_1} \ldots s_{O1_{l(O1)}} \quad \rightarrow s_{O1_{l(O1+1)}}$

...

$O_I: \quad s_{OI_1} \ldots s_{OI_{l(OI)}} \quad \rightarrow s_{OI_{l(OI+1)}}$

RS:

$R_1: \quad s_{R1_1} \ldots s_{R1_{l(R1)}}$

...

$R_J: \quad s_{RJ_1} \ldots s_{RJ_{l(RJ)}}$

$\mathrm{VARF}_{\mathrm{SIG}_{SS}}$:

$\mathrm{VAR}_{s_1} = \{x_{1_1}, \ldots, x_{1_{l(s1)}}\}$

...

$\mathrm{VAR}_{s_n} = \{x_{n_1}, \ldots, x_{n_{l(sn)}}\}$

$\mathrm{REG}_{\mathrm{SIG}_{SS}}$:

$\forall x_1, x_2: \ldots R_1(x_1, .., x_a) \rightarrow R_2(x_1, \ldots, x_a)$

$\exists x_1, x_2: \ldots R_3(x_1, \ldots, x_b)$

Die intendierten Anwendungen prädikatenlogischer Spezifikationen sind unterschiedlichster Art. Aufgrund ihrer formalen Präzision sind prädikatenlogische Spezifikationen insbesondere in solchen Anwendungsbereichen gefragt, in denen Mehrdeutigkeiten unerwünscht sind. Beispielsweise kommt hierfür das Teilgebiet des *Requirements-Engineering* im Rahmen der Software-Entwicklung in Frage.[1)] Beim Requirements-Engineering werden sowohl *funktionale* als auch *nicht-funktionale* Anforderungen spezifiziert, die eine zu konstruierende Software zu erfüllen hat. Aufgrund ihres streng formalen Aufbaus und ihrer daraus resultierenden Nähe zu Programmiersprachen werden prädikatenlogische Spezifikationen oft im Rahmen der Anforderungsanalyse bei der Software-Konstruktion herangezogen. Prädikatenlogische Spezifikationen sind nämlich grundsätzlich für Zwecke der computergestützten Anforderungsanalyse geeignet. Einzelne „Spielarten" prädikatenlogischer Spezifikationen – wie z.B. die algebraische Gleichungslogik – wurden bereits in praktischen Szenarien verwendet.

Allerdings wird durch die strenge formale Orientierung an der Prädikatenlogik eine Analyse von Anforderungen für Menschen erschwert. Zum Verständnis von prädikatenlogischen Spezifikationen wird nämlich zum einen das Verständnis der logischen Symbole, mit denen die Elemente der Menge $\mathrm{REG}_{\mathrm{SIG}_{PL}}$ konstruiert sind, vorausgesetzt. Dieses Charakteristikum der Prädikatenlogik trifft allerdings auch auf nahezu jede Sprache

1) Vgl. KREWOWSKI (1991), S. 127 ff.

zu, die beim Adressaten einer Anforderungsanalyse ein Verständnis für bestimmte Formulierungsprimitive voraussetzt, um Anforderungen auswerten zu können. Zwar ist die formale Semantik der logischen Symbole in den Regeln zur Auswertung sortierter Formeln präzise festgelegt, so dass keine Formel, ob atomar oder mittels logischer Symbole zusammengesetzt, mehrdeutig interpretiert werden kann. Mit der induktiven Grammatik zur Konstruktion von sortierten Formeln können allerdings Anforderungen konstruiert werden, die die Informationsverarbeitungskapazitäten von Adressaten der Anforderungsanalyse zur Auswertung von Formeln übersteigen. Für solche Anforderungen wird in prädikatenlogischen Spezifikationen keine Hilfestellung angeboten, die menschlichen Adressaten das Verständnis erleichtert.

Zum anderen wird für das Verständnis von prädikatenlogischen Spezifikationen und ihrer formalen Semantik vorausgesetzt, dass Adressaten von Anforderungsanalysen die formalen Objekte aus den sortenspezifischen Objektmengen als Repräsentationen realer Objekte auffassen können. Für sortierte Spezifikationen sind allerdings bislang nur formale Semantiken vorgesehen. In formalen Semantiken wird die Bedeutung von Ausdrücken und Sorten in einer spezifischen Struktur durch die jeweiligen Extensionen angegeben. Demnach ist die Bedeutung einer Sorte die sortenspezifische Objektmenge, die die Sorte interpretiert. Problematisch wird dies u.a. dann, wenn z.B. die jeweilige sortenspezifische Objektmenge leer ist.[1] Bei zeitlich veränderlichen Extensionen für Sorten kann es u.U. dazu kommen, dass eine Sorte Bedeutungsschwankungen unterliegt, weil die Extension, in Abhängigkeit von dem Zeitpunkt, in dem sie betrachtet wird, variiert. Beispielsweise würde eine Sorte „US-Präsident" – in Abhängigkeit von dem Zeitpunkt, der unterstellt wird, – eine unterschiedliche Extension aufweisen. Bei Sorten, deren Extension grundsätzlich leer sein muss, weil die formalen Objekte aus der Extension als Repräsentationen realer Objekte aufgefasst werden, wird die Situation noch weiter verschärft. Beispielsweise kann eine Sorte „Einhorn" – mit nahe liegender Interpretation – in einer formalen Semantik nicht mit einer Bedeutung versehen werden, wenn in der Extension nur formale Objekte zugelassen werden, die realweltliche Objekte repräsentieren.[2]

2.2.2 Multimengen

In einer *Multimenge*[3] wird das mehrfache Vorkommen desselben formalen Objektes als Element zugelassen.[1] Dies geschieht, indem jedem Element a einer *Trägermenge* A ei-

---

1) Dies gilt nur unter der Voraussetzung, dass leere sortenspezifische Objektmengen nicht notwendigerweise ausgeschlossen werden.

2) Vgl. CARSTENSEN ET AL. (2001), S. 253.

3) Zum Konzept der *Multimengen* vgl. GENRICH (1987), S. 214 ff.; GIRAULT/VALK (2003), S. 43; KORCZYNSKI ET AL. (1990), S. 37 ff.; REISIG (1991), S. 150 ff.; ZELEWSKI (1995), Bd. 4, S. 256. Vgl. KIFER ET AL. (1995), S. 820 f. für eine Umsetzung von Multimengen in F-Logic.

ne natürliche Zahl $n_a \in \mathbb{N}$ zugeordnet wird, die anzeigt, wie oft das Element a in der Multimenge enthalten ist. n_a wird als *Multiplizität* des jeweiligen Elements a bezeichnet. Während in der konventionellen Mengentheorie die bedeutendsten Beziehungen, die zwischen Elementen und Mengen existieren können, die „Element von"-Beziehung ($\in$) und deren Negation ($\notin$) sind, wird zwischen Elementen und Multimengen von einer „n-fachen Element von"-Beziehung ausgegangen.

Für eine beliebige Menge A ist eine Multimenge mult mit A als Trägermenge durch die Funktion

$$\text{mult}: A \rightarrow \mathbb{N}$$

definiert. Sie wird als *Multimenge über der Trägermenge A* bezeichnet. Der Funktionswert mult(a) für ein Element $a \in A$ aus der Menge A gibt dessen *Multiplizität* wieder. Die *Kardinalität* einer Multimenge mult: $A \rightarrow \mathbb{N}$ wird als #(mult) mit

$$\#(\text{mult}) = \Sigma_{a \in A}\, \text{mult}(a)$$

angegeben.[2)]

Die Menge MULT(A) umfasst *alle* Multimengen über der Trägermenge A. In jeder Multimenge $\text{mult} \in \text{MULT}(A)$ über der Trägermenge A wird jedem formalen Objekt $a \in A$ eine Multiplizität mult(a) zugeordnet. Bei einer Multimenge $\text{mult} \in \text{MULT}(A)$ bezeichnet mult(a)=0, dass a nicht in der Multimenge mult enthalten ist. Umgekehrt gilt $\text{mult}(a) \geq 1$, wenn a mindestens einmal in der Multimenge mult enthalten ist. Mit $\varnothing_M$ wird die *leere Multimenge* bezeichnet. Es gilt in diesem Fall für alle $a \in A$: $\varnothing_M(a)=0$ und somit auch $\#(\varnothing_M) = 0$. Eine konventionelle Menge A kann entsprechend auch als Multimenge mit den Funktionswerten mult(a)=1 für alle $a \in A$ aufgefasst werden.

Für den Vergleich von zwei Multimengen $\text{mult}_1, \text{mult}_2 \in \text{MULT}(A)$ werden aus der konventionellen Mengentheorie bekannte Vergleichsrelationen übertragen. Demnach gelten für die Gleichheit und für Teilmengenbeziehungen zwischen Multimengen:

- $\text{mult}_1 = \text{mult}_2 \;:\Leftrightarrow\; \forall a \in A:\; \text{mult}_1(a) = \text{mult}_2(a)$,
- $\text{mult}_1 \subseteq \text{mult}_2 \;:\Leftrightarrow\; \forall a \in A:\; \text{mult}_1(a) \leq \text{mult}_2(a)$ und
- $\text{mult}_1 \subset \text{mult}_2 \;:\Leftrightarrow\; \forall a \in A:\; \text{mult}_1(a) < \text{mult}_2(a)$.

Analog zu Vergleichsrelationen werden Operationen auf Mengen kanonisch zu Operationen auf Multimengen übertragen. Vereinigung, Durchschnitt und Differenzen zweier Multimengen $\text{mult}_1, \text{mult}_2 \in \text{MULT}(A)$ über einer Trägermenge A werden durch Operati-

1) Das mehrfache Vorkommen von Elementen in Mengen wurde bereits früher für Mengenfamilien zugelassen. Dort erfolgt allerdings eine Differenzierung der Mitgliedsmengen durch die Indexierung der Mitglieder. Dies ist zwar für Multimengen grundsätzlich nicht vorgesehen, hat aber keinen gegenseitigen Ausschluss der beiden Konzepte zur Folge, da jede Mengenfamilie als eine Multimenge aufgefasst werden kann. Auf diesen Aspekt von Multimengen wird weiter unten eingegangen.

2) Vgl. GIRAULT/VALK (2003), S. 43; JENSEN (1996), S. 68; KLEINJOHANN (1993), S. 56; NÜTTGENS/RUMP (2002), S. 67; WEBER (2002), S. 9.

onen auf Multiplizitäten der betroffenen Multimengen für jedes Element $a \in A$ aus der gemeinsam zugrunde liegenden Trägermenge A wie folgt definiert:

- $mult_1 \cup mult_2 = mult_3 \quad :\Leftrightarrow \quad \forall a \in A: \quad mult_3(a) = \max(mult_1(a), mult_2(a))$,
- $mult_1 \cap mult_2 = mult_3 \quad :\Leftrightarrow \quad \forall a \in A: \quad mult_3(a) = \min(mult_1(a), mult_2(a))$ und
- $mult_1 \setminus mult_2 = mult_3 \quad :\Leftrightarrow \quad \forall a \in A: \quad mult_1(a) \geq mult_2(a) \rightarrow mult_3(a) = mult_1(a) - mult_2(a))$.

Darüber hinaus ist die *komponentweise Addition* von Multimengen wie folgt definiert:

- $mult_1 \oplus mult_2 = mult_3 \quad :\Leftrightarrow \quad \forall a \in A: \quad mult_3(a) = mult_1(a) + mult_2(a)$.

Die komponentenweise Addition $\oplus$ unterscheidet sich von der Vereinigung $\cup$ dadurch, dass bei ihrer Anwendung nicht die bezüglich eines Elements a größere Multiplizität aus den beiden Multimengen, sondern die Summe der multimengenspezifischen Multiplizitäten angesetzt wird.

Um Multimengen auszudrücken, bieten sich grundsätzlich mehrere Möglichkeiten an.[1)] In der am weitesten verbreiteten Ausdrucksvariante werden Multimengen als *formale Summen* notiert.[2)] Eine Multimenge $mult \in MULT(A)$ mit der Trägermenge $A=\{a_1,...,a_n\}$ wird bei dieser Ausdrucksvariante als formale Summe der Form

$$\begin{aligned} mult &= mult(a_1)a_1 + ... + mult(a_n)a_n \\ &= \Sigma_{i=1,...,n}\, mult(a_i)a_i \end{aligned}$$

angegeben. Dabei werden nur – mit Ausnahme der leeren Menge $\varnothing$ – Multiplizitäten mit $mult(a_i)>1$ aufgeführt. Die leere Multimenge $\varnothing_M$ wird als degenerierte formale Summe als Skalar 0 notiert.

Im Folgenden werden Multimengen und ihre Notation als formale Summen mittels eines *Beispiels* verdeutlicht. Für die Trägermenge $A=\{a,b,c\}$ seien die beiden Multimengen

---

1) Vgl. ZELEWSKI (1995), Bd. 4, S. 259 ff.
Über die dort erwähnten Alternativen zum Ausdrücken von Multimengen hinaus wird öfters in Zusammenhang mit algebraischen Petri-Netzen eine weitere Alternative verwendet; vgl. AOUMEUR (2001), S. 35; REISIG (1991), S. 143. Neben der „originären“ Signatur SIG_{SS}, die einem algebraischen Petri-Netz zugrunde liegt, wird dort zusätzlich eine „derivative“ *Multimengensignatur* $MSIG_{SS}$ konstruiert. Jeder Sorte s der originären Signatur SIG_{SS} wird in einer Multimengensignatur $MSIG_{SS}$ eine Multimengensorte ms zugewiesen. Zudem werden die semantischen Anforderungen an Multimengen in Form von Regeln formuliert, wodurch die Multimengensignatur $MSIG_{SS}$ zu einer *Multimengenspezifikation* $MSPEZ_{SS}$ ausgeweitet wird.

Für Prädikat/Transition-Netze ist eine solche Repräsentation von Multimengen auf der Basis von Multimengenspezifikationen nicht üblich. Auch für Ontologie-Netze wird nicht auf Multimengenspezifikationen zurückgegriffen. Daher wird dieser Ansatz im Folgenden nicht weiter berücksichtigt.

2) Vgl. GENRICH (1987), S. 215; GIRAULT/VALK (2003), S. 43; KORCZYNSKI ET AL. (1990), S. 37; ZELEWSKI (1995), Bd. 4, S. 263.

$mult_1$: $A \rightarrow \mathbb{N}$			und	$mult_2$: $A \rightarrow \mathbb{N}$		
	$mult_1(a)$	$= 3$			$mult_2(a)$	$= 1$
	$mult_1(b)$	$= 2$			$mult_2(b)$	$= 2$
	$mult_1(c)$	$= 2$			$mult_2(c)$	$= 1$

gegeben. Sie können äquivalent als formale Summen $mult_1$=3a+2b+2c bzw. $mult_2$=a+2b+c notiert werden. Die Summe $mult_1 \oplus mult_2$ der beiden Multimengen lautet 4a+4b+3c. Die Differenz $mult_1 \backslash mult_2$ der beiden Multimengen lautet 2a+c.

3 Bausteine des integrativen Modellierungskonzepts

3.1 Ontologien

3.1.1 Überblick über das Ontologieverständnis

3.1.1.1 Informales Ontologieverständnis

Der Begriff *Ontologie* wird bisweilen mit teilweise sehr unterschiedlichen Verständnissen verbunden. Zumeist wird allerdings das jeweilige Verständnis natürlichsprachlich angedeutet und in den „günstigsten“ Fällen mit exemplarischen Formalisierungen verdeutlicht. Eine solche Vorgehensweise birgt die Gefahr, den Begriff zu „verwässern“ und gegenüber alternativen Konzeptionen keine klare Trennschärfe zu bieten. Um für die vorliegende Arbeit eine präzise Argumentationsgrundlage zu schaffen, wird daher das Verständnis, das mit der Bezeichnung „Ontologie“ verbunden wird, *formal* festgelegt. Um dennoch eine Diskussionsgrundlage auf der Basis einer informalen Definition zu erhalten, wird im Folgenden ein Definitionsvorschlag für den Begriff *Ontologie* übernommen, der mit der formalen Präzisierung kompatibel ist.

Bezüglich der informalen Definitionsvorschläge für Ontologien kann auf eine Vielzahl von Arbeiten zurückgegriffen werden.[1)] Der Definitionsvorschlag von GRUBER ist jedoch in der Ontologie-Forschung am weitesten verbreitet. Nach GRUBER kann eine Ontologie definiert werden als:

„an explicit specification of a conceptualization“[2)].

Der Begriff *conceptualization*, auf den sich die Definition bezieht, wird von GRUBER als eine abstrakte, vereinfachte Sichtweise auf Phänomene der Welt beschrieben, die von einem *Subjekt* aus einem bestimmten *Zweck* heraus eingeschlagen wird.[3)] Dabei kann vorausgesetzt werden, dass jede Konzeptualisierung auch immer eine subjekt- und zweckabhängig konstruierte *begriffliche* Struktur ist.[4)]

1) Für Überblicke über alternative Definitionen von Ontologien vgl. GUARINO (1997), S. 295 ff.; VAN HEIJST ET AL. (1997), S. 191 ff.

2) GRUBER (1992A), S. 1; GRUBER (1993), S. 199.

3) GRUBER (1993), S. 199. Der Subjekt- und Zweckbezug von Ontologien wird darüber hinaus implizit auch in anderen Quellen angesprochen. Vgl. z.B. PIRLEIN/STUDER (1995), S. 946: „ ... an ontologically is typically not totally task-independent since in all practical cases it can be assumed that the knowledge engineer constructed it with at least a very general task in mind.“; O´LEARY spricht in diesem Zusammenhang sogar von einem „politischen Prozess“ bei der Konstruktion von Ontologien; vgl. O´LEARY (1997), S. 328.

4) Auf die Notwendigkeit des Sprachbezugs jeder Konzeptualisierung wird in Abschnitt 3.1.3.2.1.1 näher eingegangen.

Die Definition lässt es jedoch offen, ob eine interne und somit natürlichsprachliche oder eine externe und somit nicht notwendigerweise formalsprachliche Konzeptualisierung Ausgangspunkt für die Konstruktion einer Ontologie ist.[1] Im Fall der internen und somit natürlichsprachlichen Konzeptualisierung wird eine Ontologie unmittelbar ohne eine zwischenstufige Externalisierung konstruiert. Die natürlichsprachlich vorliegenden internen Konzeptualisierungen der beteiligten Akteure werden direkt in einer Ontologie formal rekonstruiert. Eine solche Vorgehensweise ist allerdings in analoger Weise bereits in frühesten Phasen des *Software-Engineerings* als problematisch erkannt worden. Denn eine solche Rekonstruktion von internen und somit natürlichsprachlichen Konzeptualisierungen durch formalsprachliche Konstrukte scheitert oftmals an den Wissensbestandteilen, die sich einer unmittelbaren Formalisierung entziehen. Solche Wissensbestandteile sind im Rahmen des betrieblichen Wissensmanagements in den letzten Jahren zunehmend unter dem Begriff des *tacit knowledge*[2] thematisiert worden.

Gerade solche taziten Wissensbestandteile sind es allerdings in der Regel, die die Spezifität akteursspezifischen Wissens ausmachen. Um auch solches Wissen ontologiegestützt repräsentieren zu können, ist es notwendig, Methoden der *Wissensakquisition* anzuwenden, mit deren Hilfe das Wissen eines Akteurs *erhoben* und *dokumentiert* werden kann.[3] Das Ergebnis der Wissensakquisition – in der Regel handelt es sich hierbei um semi-formale Darstellungen – kann im Anschluss in eine Ontologie überführt werden. Insofern wird von der zweiten Alternative ausgegangen, derzufolge Ontologien Spezifikationen von externen Konzeptualisierungen sind.[4]

---

1) Auf die Unterscheidung zwischen interner und externer Konzeptualisierung wurde bereits früher hingewiesen; vgl. Abschnitt 2.1.1. Die interne Konzeptualisierung entspricht einem mentalen Modell eines Akteurs. Die externe Konzeptualisierung entspricht einer kommunizierbaren Form des mentalen Modells.

2) Zum „tacit knowledge"-Phänomen vgl. NEUWEG (2001), S. 15 ff.; POLANYI (1985), S. 13 ff.

3) Der Verfasser hat an anderer Stelle Methoden zur Akquisition von Wissen – insbesondere von Wissen über akteursspezifische Kompetenzen – vorgestellt und bewertet; vgl. ALAN (2002), S. 26 ff. Vgl. darüber hinaus zur Wissensakquisition KARBACH/LINSTER (1990), S. 9 ff.

4) Trotz der oftmals fehlenden unmittelbaren Spezifizierbarkeit interner Konzeptualisierungen wird oftmals von der zwischenstufigen Explikation abstrahiert. Auch für die folgenden Ausführungen wird vorausgesetzt, dass die objektsprachlichen Komponenten aus Ontologien den Komponenten der internen Konzeptualisierung zugeordnet werden können. Insofern wird zwar eine Zwischenstufe vorausgesetzt, die zwischen der internen Konzeptualisierung und der Ontologie liegt, jedoch verbleibt die Existenz dieser Stufe lediglich *implizit*. Auf die Prozesse der Überführung interner Konzeptualisierungen in semi-formale Fragmente der Zwischenstufe wird nicht mehr eingegangen. Denn für die folgenden Ausführungen sind in erster Linie die *formalen* Aspekte von Ontologien von Bedeutung, um hierauf aufbauend eine operationale Semantik für ontologiegestützte Modelle vorstellen zu können. Somit wird auch von der Überführung der externen Konzeptualisierungen in Ontologien abgesehen.

Mit *specification* wird verdeutlicht, dass die Bestimmung der Begriffsstruktur in einer *formalen* Sprache zu erfolgen hat.[1] Somit werden natürliche und semi-formale Sprachen für die Ontologie-Konstruktion ausgeschlossen. Ontologien sind daher stets formalsprachliche Artefakte. Zudem wird durch das Attribut *explicit* eingefordert, dass die Festlegung der Begriffsstruktur auf keine Weise implizite Aspekte aufweisen darf. Sämtliche intendierten Festlegungen müssen demnach auch als solche formuliert werden. Ansonsten werden sie nicht berücksichtigt.

Die Definition von GRUBER ist zwar als Ausgangspunkt dafür, was unter dem Begriff *Ontologie* zu verstehen ist, sehr hilfreich. Sie wird jedoch aus mehreren Gründen der vorliegenden Arbeit nicht in ihrer ursprünglichen Form als Arbeitsdefinition zu Grunde gelegt.

Erstens wird die Definition durch ihre Präsuppositionen einer methodischen Inkonsistenz ausgesetzt. Einerseits wird nämlich *in* der Definition eine Explizitheitsprämisse für Ontologien formuliert. Demnach dürfen Ontologien als wissensrepräsentierende Artefakte keine „unausgesprochenen" Bestandteile aufweisen. Andererseits werden aber *für* die Definition einige Voraussetzungen getroffen, die erst durch weitergehende Erläuterungen vermittelt werden können. So wird für die Begriffe „specification" und „conceptualization" ein Vorverständnis beim Leser vorausgesetzt, das nicht notwendigerweise gegeben sein muss.

Der zweite Grund, aus dem heraus die GRUBER-Definition der vorliegenden Arbeit nicht als Arbeitsdefinition zugrunde gelegt wird, wurde allem von GUARINO thematisiert.[2] Er handelt sich hierbei darum, dass der Begriff der *Konzeptualisierung* bei GRUBER mit einer *extensionalen* Erfahrung[3] eines Realitätsausschnitts einhergeht. Entsprechend dieser extensionalen Sichtweise korreliert eine Konzeptualisierung auch stets mit einem bestimmten *Zustand* („state of affairs") des betrachteten Realitätsausschnitts. Die semanti-

---

1) Einerseits wird hierbei vorausgesetzt, dass sich die Bezeichnung *Spezifikation* aufgrund von Arbeiten aus dem Umfeld des *Requirements-Engineerings* stets auf *formalsprachliche* Artefakte bezieht. Andererseits geht die Forderung nach Formalsprachlichkeit aus den späteren Anmerkungen GRUBERS hervor. Beispielsweise spricht GRUBER im unmittelbaren Anschluss an die Definition von „objects, and the formalized relationships among them" und von „formal axioms" (GRUBER (1993), S. 199).

2) Vgl. GUARINO (1997), S. 294 ff.

3) Die extensionale Erfahrung eines Realitätsausschnitts spiegelt sich darin wieder, dass nach GRUBER in einer Ontologie u.a. auch *Individualbegriffe* vorkommen können, mit denen auf bestimmte Erfahrungsobjekte referenziert wird. Individualbegriffen entsprechen in ontologiegestützten Modellen solche formalen Objekte („Instanzen"), die den Extensionen von Konzepten zugeordnet werden. GRUBER führt Instanzen („individuals") als Komponenten von Ontologien auf; vgl. GRUBER (1992), S. 1. Im Gegensatz zu Instanzen handelt es sich bei Konzepten zumeist um *Gattungsbegriffe*, mit denen nicht auf bestimmte Erfahrungsobjekte, sondern auf deren Gattungen referenziert wird. Für die vorliegende Arbeit werden Instanzen aus Ontologien ausgeschlossen. Daher wird auch im weiteren Verlauf von der undifferenzierten Verwendung des Begriffs „Konzeptualisierung" Abstand genommen.

sche Ausdrucksmächtigkeit von Ontologien rührt nach Ansicht des Verfassers hingegen daher, dass nicht nur extensionale, sondern auch *intensionale* Aspekte formal erfasst werden.[1)] Ontologien sind – im Gegensatz zu Modellen eines betrachteten Realitätsausschnitts – nicht an einen bestimmten Realitätsausschnitt gekoppelt. Vielmehr werden in Ontologien die *sprachlichen Ausdrucksmittel* spezifiziert, mittels derer ein Realitätsausschnitt repräsentiert werden kann. Insofern sind Ontologien auf die Spezifikation von *Denkmöglichkeiten* über einen Realitätsabschnitt bezogen.

Ein weiterer Kritikpunkt an der GRUBER-Definition liegt darin begründet, dass sie den ontologischen Status der Realität offen lässt, auf den sich eine Ontologie bezieht. Zwar wird durch den Subjekt- und Zweckbezug jeder Konzeptualisierung verdeutlicht, dass es sich hierbei immer um aktive Konstruktionsleistungen von erkennenden Subjekten handelt. Allerdings geht dies aus der Definition selbst nicht unmittelbar hervor. Darüber hinaus wird in der Regel für Ontologien eingefordert, dass die Konstruktion von Ontologien mit einem Prozess der Einigung *mehrerer* Akteure über die spezifizierten Begrifflichkeiten einhergeht.[2)] Auch dieser Aspekt von Ontologien ist in der Definition nicht zu erkennen.

Den folgenden Ausführungen wird eine Erweiterung der GRUBER-Definition durch ZELEWSKI zugrunde gelegt: „Eine Ontologie ist eine explizite und formalsprachliche Spezifikation derjenigen sprachlichen Ausdrucksmittel (für die Konstruktion repräsentationaler Modelle), die nach Maßgabe einer von mehreren Akteuren gemeinsam verwendeten Konzeptualisierung von realen Phänomenen, die in einem subjekt- und zweckabhängig eingegrenzten Realitätsausschnitt als wahrnehmbar oder vorstellbar gelten und für die Kommunikation zwischen den o.a. Akteuren benutzt oder benötigt werden, für „sinnvoll“ erachtet werden“[3)].

Die Arbeitsdefinition weist einerseits den Vorteil auf, den Anschluss an die mittlerweile etablierte Sichtweise auf Ontologien nicht zu verlieren. Andererseits werden jedoch die unannehmbaren Defizite der ursprünglichen GRUBER-Definition behoben. Die wichtigste Erweiterung liegt darin, die ontologische Spezifikation nicht mehr auf eine Konzeptualisierung zu beziehen, sondern auf die sprachlichen Ausdrucksmittel, mit denen Konzeptualisierungen konstruiert werden können. Explizit wird hierbei auf die „Sinnhaftigkeit“ der sprachlichen Ausdrucksmittel in der Betrachtung der Akteure Bezug genommen. Dadurch werden beispielsweise Relationen ausgeschlossen, mit deren Hilfe

---

1) Vgl. GUARINO/GIARETTA (1995), S. 28; GUARINO (1997), S. 296. In weiteren Arbeiten hat GUARINO sein Verständnis von Ontologien unter Rückgriff auf die Modallogik präzisiert; vgl. GUARINO (1998), S. 4 ff. Darüber hinaus findet sich der Bezug von Ontologien zur intensionalen Semantik sprachlicher Ausdrücke in zahlreichen Anmerkungen aus Arbeiten zu Ontologien wieder. Vgl. hierzu z.B. NOY ET AL. (2001), S. 62: („ ... let us create intensional class definitions ... “); VAN HEIJST ET AL. (1997), S. 183: („ ... ontologies – intensional descriptions of the domain knowledge in some field.“).

2) Vgl. STUDER ET AL. (1998), S. 184; SURE (2003), S. 23.

3) ZELEWSKI (2005) [Hervorhebungen im Original hier unterlassen].

eine „sinnlose“ Beziehung zwischen Objekten ausgedrückt werden könnte. Der Ausschluss von ontologiegestützten Modellen, in denen „sinnlose“ Aussagen enthalten sind, erfolgt dadurch, dass Ontologien nunmehr auf „wahrnehmbare oder vorstellbare“ und nicht auf bestimmte Konzeptualisierungen bezogen werden.

Der Einschränkung auf „sinnvoll“ erachtete (natürlich-)sprachliche Ausdrucksmittel auf objektsprachlicher Ebene entspricht der Arbeitsdefinition zufolge eine Einschränkung auf formalsprachliche Ausdrucksmittel auf metasprachlicher Ebene. Demzufolge werden als Ontologien lediglich formalsprachliche Artefakte zugelassen. Natürlichsprachliche oder semi-formale Artefakte werden hingegen als Ontologien ausgeschlossen.[1)] Zudem wird in der Arbeitsdefinition ein Schwerpunkt auf die *Kommunikation* von Akteuren gelegt. Jedoch sind formalsprachliche Artefakte als Kommunikationsmedium für menschliche Akteure in der Regel nicht geeignet. Dies ist nur ein vermeintlicher Widerspruch, da nicht eingefordert wird, die Ontologie selbst als formalsprachliches Medium für die Kommunikation zu verwenden. Dies bietet sich in erster Linie für die Kommunikation zwischen maschinellen Akteuren an. Die Formalsprachlichkeit von Ontologien erstreckt sich vielmehr auf die formale Rekonstruktion der natürlichen Begriffswelt. Dies ist insofern erstrebenswert, als dass auch als Kommunikationsgrundlage für maschinelle Akteure Begriffe aus einer natürlichen Sprache verwendet werden können. Somit kann die Kommunikation maschineller Akteure an den natürlichen Sprachgebrauch angelehnt werden. Darüber hinaus können Ontologien auch für die Kommunikation zwischen menschlichen Akteuren verwendet werden. Beispielsweise sind Instrumente denkbar, die sich im Hintergrund einer Ontologie bedienen, um die natürlichsprachliche Kommunikation menschlicher Akteure mit unterschiedlichem Sprachhintergrund zu harmonisieren.

---

1) Eine solche Einschränkung hat weit reichende Konsequenzen, so dass beispielsweise Arbeiten zu „UML-Ontologien“ (vgl. z.B. BACLAWSKI ET AL. (2002), S. 145 ff.; CRANEFIELD ET AL. (2003), S. 55 ff.) die Argumentationsgrundlage entzogen wird. Für die Konstruktion von Ontologien werden in erster Linie *UML-Klassendiagramme* diskutiert, die allerdings keine formale Semantik aufweisen. U.a. von CRANEFIELD wurde allerdings dieses Defizit von UML für die Ontologiekonstruktion in KOGUT ET AL. (2002), S. 61 angesprochen und mit der Hoffnung verknüpft, in naher Zukunft eine formale Semantik für UML-Klassendiagramme zu definieren.

3.1.1.2 Formales Ontologieverständnis

Für die *formale* Präzisierung des Verständnisses von Ontologien wird auf den formalen Rahmen zurückgegriffen, der in den vorherigen Abschnitten entfaltet wurde. Hierzu werden Ontologien in den folgenden Abschnitten als eine Erweiterung sortierter Spezifikation vorgestellt.[1)]

Sortierte Spezifikationen wurden in Abschnitt 2.2.1.3 als eine Ausprägungsform prädikatenlogischer Spezifikationen vorgestellt. Erstgenannte wurden wiederum als Erweiterung sortierter Signaturen um jeweils eine Menge von sortierten Formeln mit Anforderungscharakter vorgestellt.

Ontologien basieren hingegen auf *ontologischen Signaturen*. In einer ontologischen Signatur SIG_{OS} werden die objektsprachlichen Ausdrucksmittel spezifiziert, die für die Konstruktion von Ausdrücken benötigt werden. So werden auch alle Elemente der Regelkomponente einer Ontologie als Ausdrücke über einer ontologischen Signatur SIG_{OS} eingeführt.

Dadurch, dass das hier vorgelegte Verständnis über Ontologien auf bekannten prädikatenlogischen Spezifikationen basiert, wird die Anschlussfähigkeit des integrativen Modellierungskonzepts an bereits vorliegende Arbeiten gewährleistet. Somit lassen sich Erkenntnisse, die im Bereich der prädikatenlogischen Spezifikationen vorliegen, möglicherweise analog auf Ontologien übertragen.

Im Vergleich zu prädikatenlogischen Spezifikationen erfolgen durch Ontologien einige wesentliche Erweiterungen. Mit den Erweiterungen wird das Ziel verfolgt, Defizite der konventionellen und sortierten Prädikatenlogik zu überbrücken.[2)] Es handelt sich hierbei in erster Linie um die Defizite, die sich aus der vorwiegenden Berücksichtigung der *extensionalen* Semantik objektsprachlicher Konstrukte ergeben. Mit Ontologien werden verstärkt auch *intensionale* Aspekte objektsprachlicher Konstrukte berücksichtigt.

Die Extension eines deskriptiven Symbols wurde bereits als die Menge seiner *Instanzen* vorgestellt. Im Fall von Operations- und Relationssymbolen handelt es sich bei den Instanzen um Individuentupel, die in den Extensionen der Operations- und Relationssym-

1) Insofern kann das hier vorgelegte formale Verständnis für Ontologien als „Meta-Ontologie“ charakterisiert werden. Im Rahmen dieser Meta-Ontologie werden die Komponenten von Ontologien aufgeführt. Auch für diese meta-ontologische Spezifikation gelten die Ausführungen im Rahmen des wissenschaftstheoretischen Rahmens. Insofern kann auch hier von eine subjektbezogenen Konstruktion ausgegangen werden. Es lassen sich ohne Zweifel alternative Meta-Ontologien vorstellen, denen zufolge Ontologien eine alternative Struktur aufweisen. Der Verfasser weist mit der Vorlage „seiner“ Meta-Ontologie keinesfalls alternative Konstruktionen zurück. Allerdings ist es für die Bearbeitung der Problemstellung notwendig, eindeutig Stellung zu beziehen, von welcher Konstruktion ausgegangen wird.

2) Auf eine Evaluation der konventionellen und sortierten Prädikatenlogik mit Hilfe des Anforderungskatalogs für die statische Struktur wird verzichtet, da sie nicht im Erkenntnisinteresse der vorliegenden Arbeit liegt.

bole enthalten sind. Für Operationssymbole konstituieren die Individuentupel eine Operation. Für Relationssymbole wird durch die Individuentupel jeweils eine Relation konstituiert. Die Instanzen von Sorten sind Individuen, die in den sortenspezifischen Objektmengen enthalten sind.

Die *Intension* eines deskriptivn Symbols umfasst hingegen die *Merkmale*, die auf das Symbol zutreffen und anhand derer bestimmt werden kann, welche Objekte Instanzen des Symbols sind.[1)] Somit ist die Extension eines sprachlichen Konstrukts seiner Intension stets untergeordnet.[2)] Beispielsweise entspricht die Intension einer Sorte $s \in S$ der Menge aller Merkmale, die ein formales Objekt ob_u in der sortenspezifischen Objektmenge OB_s erfüllen muss. Die Intension eines Relationssymbols R_j entspricht hingegen den Merkmalen, denen ein Objekttupel $(ob_1,...,ob_n)$ entsprechen muss, um in der Extension r_j des Relationssymbols R_j enthalten zu sein. Allerdings werden in prädikatenlogi-

1) Vgl. CANN (1993), S. 267; JOHNSON-LAIRD (1983), S. 195 ff.; ORTNER (1997), S. 30; SOWA (2000), S. 99.

2) Die Unterordnung der Extension eines sprachlichen Konstrukts gegenüber seiner Intension tritt insbesondere bei der *Modallogik* zu Tage; vgl. CANN (1993), S. 263 ff.; LÖBNER (2003), S. 351 ff.; VON KUTSCHERA (1976), S. 18. Im Rahmen der Modallogik wird die *Intension* eines Relationssymbols R_j mit der Menge aller seiner *denkbaren Extensionen* gleichgesetzt; vgl. GUARINO (1998), S. 5; VON KUTSCHERA (1976), S. 24. Die Strukturen, die für die extensionale Interpretation einer prädikatenlogischen Signatur SIG_{PL} in Frage kommen, werden hierzu jeweils mit Indizes versehen, um sie voneinander unterscheiden zu können. Aufgrund der expliziten Behandlung der Abhängigkeit der extensionalen Semantik von der intensionalen Semantik wird die Modallogik oftmals auch als *intensionale Logik* bezeichnet.

schen Signaturen keine Ausdrucksmittel zur Verfügung gestellt, um die Intensionen sprachlicher Ausdrucksmittel anzugeben.[1)]

Analog zum Aufbau prädikatenlogischer Spezifikationen handelt es sich bei Ontologien um eine Zusammensetzung bestehend aus einer *Signatur* und einer[2)] Menge von *Regeln*. Signaturen, die einer Ontologie zugrunde liegen, werden als *ontologische Signaturen* bezeichnet. Als objektsprachliche Ausdrucksmittel kommen in einer ontologischen Signatur SIG_{OS} *Konzepte*, *Operationssymbole*, *Relationssymbole* und *Variablen* in Frage. Konzepte sind die „ontologischen" Pendants zu *Sorten* aus sortierten Signaturen.[3)] Darüber hinaus unterscheiden sich Operations- und Relationssymbole aus einer ontologischen Signatur SIG_{OS} nicht von Operations- bzw. Relationssymbolen, die in einer sortierten Signatur SIG_{SS} Verwendung finden.[4)] Der extensionalen Interpretation von Ope-

---

1) Die Merkmale, die Objekte und Objekttupel aufweisen müssen, um in den Extensionen von Sorten bzw. Operations- oder Relationssymbolen enthalten zu sein, lassen sich in *notwendige* und *hinreichende* Merkmale unterscheiden. Im Rahmen der konventionellen und sortierten Prädikatenlogik können mit Hilfe der Typisierungsfunktionen lediglich *notwendige* Merkmale von Objekten und Objekttupeln angegeben werden, um entsprechend der intensionalen Semantik von deskriptiven Symbolen in deren Extensionen enthalten zu sein. So muss z.B. die Extension r_j jedes Relationssymbols R_j entsprechend zu dessen Typ $typ_{RS_{PL}}(R_j)$ aufgebaut sein. Im Fall der konventionellen Prädikatenlogik muss r_j die gleiche numerische Stelligkeit wie R_j haben. Im Fall der sortierten Prädikatenlogik muss r_j über sortenspezifischen Objektmengen entsprechend der Typisierung von R_j konstruiert sein. Dadurch werden alle Objekttupel $(ob_1,...,ob_n)$ als Instanzen von R_j mit $typ_{RS_{SS}}(R_j)=(s_1,...,s_n)$ ausgeschlossen, wenn $(ob_1,...,ob_n)\notin(OB_{s_1}\times...\times OB_{s_n})$ gilt. Mit Hilfe der Typisierungsfunktionen werden somit extensionale Interpretationen von Operations- und Relationssymbolen ausgeschlossen, wenn sie nicht typgerecht sind. Entsprechend werden durch Typisierungsfunktionen implizit die Intensionen von Operations- und Relationssymbolen berücksichtigt. Dabei werden auch intensionale Aspekte von Sorten berücksichtigt, da Sorten zur Typisierung von Operations- und Relationssymbolen verwendet werden können. Die Extension OB_s einer Sorte s kann nämlich nur solche formalen Objekte umfassen, deren Vorkommen in einem Objekttupel, das zur extensionalen Interpretation eines Operations- oder Relationssymbols verwendet wird, „sinnvoll" ist. Allerdings können darüber hinausgehende *hinreichende* Merkmale für die Zugehörigkeit eines Objektes zu der Extension eines deskriptiven Symbols weder in der konventionellen noch in der sortierten Prädikatenlogik angegeben werden.

2) Bei einer präzisieren Diktion müsste von *zwei* Mengen von Regeln gesprochen werden. In Ontologien wird nämlich sowohl formal als auch material zwischen *Inferenz*- und *Integritätsregeln* unterschieden. Eine solche Unterscheidung setzt allerdings die formale Ausarbeitung der folgenden Abschnitte voraus.

3) Die Bezeichnungen „Konzept" und „Sorte" werden sogar in einem folgenden Abschnitt als Synonyme deklariert. Allerdings bedarf es weiterführender Erläuterungen, die eine solche Deklaration erlauben. Vgl. hierzu Abschnitt 3.1.3.2.1.1.

4) Auf die Eigenarten von Operations- und Relationssymbolen aus ontologischen Signaturen wird in den Abschnitten 3.1.3.2.1.2 bzw. 3.1.3.2.1.3 näher eingegangen. Bei einer solchen erneuten Vertiefung von Operations- und Relationssymbolen wird bewusst die Gefahr redundanter Erörterungen in Kauf genommen, um den Aufbau ontologischer Signaturen und ihrer Interpretationen in „abgeschlossener" Form ohne Querverweise präsentieren zu können.

rationssymbolen aus ontologischen Signaturen entsprechen – wie im Fall von Operationssymbolen aus sortierten Signaturen auch – Operationen aus Strukturen, die jenen Signaturen zugeordnet werden können. Die Extensionen von Relationssymbolen über ontologischen Signaturen sind hingegen Relationen aus entsprechenden Strukturen.

Neben Anforderungen an die Beschaffenheit der konzeptspezifischen Objektmengen können in ontologischen Signaturen auch Anforderungen an die Beschaffenheit von Operationen und Relationen ausgedrückt werden. Im Gegensatz zu konventionellen Signaturen zeichnen sich nämlich ontologische Signaturen dadurch aus, dass sie über eine Vielzahl metasprachlicher Ausdrucksmöglichkeiten verfügen, anhand derer solche Strukturen als formale Semantik einer ontologischen Signatur ausgeschlossen werden können, die für die extensionale Interpretation der Signatur aufgrund ihrer „Sinnlosigkeit“ nicht in Frage kommen.

Dieses Prinzip ist bereits in wesentlich verkürzter Form aus konventionellen und in äquivalenter Form aus sortierten Signaturen bekannt. Um metasprachliche Anforderungen an Strukturen zu prädikatenlogischen Signaturen formulieren zu können, stehen nämlich dort die Typisierungsfunktionen $typ_{OPS_{KS}}$ und $typ_{RS_{KS}}$ – im Fall der konventionellen Prädikatenlogik – bzw. $typ_{OPS_{SS}}$ und $typ_{RS_{SS}}$ – im Fall der sortierten Prädikatenlogik – zur Verfügung. Mit Hilfe der Typisierungsfunktionen aus der Prädikatenlogik können notwendige Bedingungen formuliert werden, denen Objekttupel genügen müssen, damit sie als Instanzen von Operations- oder Relationssymbolen in Frage kommen.

Analog hierzu werden auch in ontologischen Signaturen metasprachliche Funktionen für die Typisierung von Operations- und Relationssymbolen verwendet. Darüber hinaus können in ontologischen Signaturen auch solche metasprachlichen Ausdrücke konstruiert werden, die *hinreichenden* Bedingungen für die Zugehörigkeit von Objekten zu den Extensionen von Konzepten entsprechen.

3.1.2 Syntaktische Aspekte von Ontologien

3.1.2.1 Ontologische Signaturen

Das Alphabet $ALPH_{OS}$ zur Konstruktion einer Ontologie geht aus dem Alphabet $ALPH_{KS}$ für die konventionelle Prädikatenlogik dadurch hervor, dass es um eine Menge K von *Konzepten* erweitert wird:

$$ALPH_{OS} = ALPH_{KS} \cup K.$$

Das formalsprachliche Alphabet $ALPH_{OS}$ umfasst demnach alle logischen Symbole, die zur Konstruktion von Formeln benötigt werden. Darüber hinaus sind in $ALPH_{OS}$ alle deskriptiven Symbole aus $ALPH_{KS}$ enthalten. Hierzu gehören neben Operations- und Relationssymbolen auch Variablen. Auch Konzepte werden fortan zu der Menge aller deskriptiven Symbole gezählt. Schließlich umfasst das formalsprachliche Alphabet $ALPH_{OS}$ Hilfszeichen, die für die Konstruktion von Aussagen benötigt werden.

Die Zulässigkeit von Aussagen, die mit den Symbolen aus dem formalsprachlichen Alphabet $ALPH_{OS}$ konstruiert werden, ist immer in Bezug auf eine ontologische Signatur SIG_{OS} definiert. In einer ontologischen Signatur[1)] SIG_{OS} werden Typisierungen von Operations- und Relationssymbolen vorgenommen, die für die Überprüfung der Zulässigkeit von Aussagen benötigt werden. Darüber hinaus umfasst eine ontologische Signatur SIG_{OS} weitere metasprachliche Konstrukte.

Ontologische Signaturen sind notwendiger Bestandteil jeder Ontologie. Vereinfachend werden ontologische Signaturen daher auch als *Ontologien i.e.S.* bezeichnet. Hinreichende Komponenten von Ontologien sind hingegen *Regeln*, die über einer ontologischen Signatur SIG_{OS} konstruiert sind. Auf die Erweiterung ontologischer Signaturen um Regeln und *Ontologien i.w.S.* wird in einem späteren Abschnitt näher eingegangen.[2)]

1) Die Begriffe *ontologische Signatur* und *Ontologie-Signatur* werden synonym verwendet.

2) Vgl. Abschnitt 3.1.4.1.

Eine ontologische Signatur SIG_{OS} ist definiert als das 13-Tupel

$$SIG_{OS}=(K,MEN,ALPH_{META},\sqsubseteq,\circeq,\curlyvee,OPS,typ_{OPS_{OS}},RS,typ_{RS_{OS}},VARF_{SIG_{OS}},bezf,deff).$$

Die Komponenten einer ontologischen Signatur SIG_{OS} sind:

(1.) Eine Menge $K=\{k_1,...,k_N\}$ von *Konzepten* mit $n=1,...,N$, $N\in\mathbb{N}_+$,
$K = K_{EW} \cup K_{MW} \cup \{\top,\bot\}$
mit $K_{EW} \cap K_{MW} = \varnothing$.

(2.) eine *Mengenfunktion* MEN: $K_{EW} \rightarrow K_{MW}$,

(3.) eine Menge $ALPH_{META}$ *meta-* und *natürlichsprachlicher Zeichen*
$ALPH_{META}=\{\texttt{0,1,...,9,a,b,...,z,A,B,...,Z,,,;,.,:,+,-,?,!,\$,\%;/,(,),=,\ ,}\}$,

(4.) eine *Subkonzeptrelation* $\sqsubseteq \subseteq (K \times K)$
mit (4.1.) $\forall k\in K$: $(k \sqsubseteq k)$,
(4.2.) $\forall k_1,k_2\in K$: $((k_1 \sqsubseteq k_2) \wedge (k_2 \sqsubseteq k_1)) \rightarrow (k_1 = k_2)$,
(4.3.) $\forall k_1,k_2,k_3\in K$: $((k_1 \sqsubseteq k_2) \wedge (k_2 \sqsubseteq k_3)) \rightarrow (k_1 \sqsubseteq k_3)$,
(4.4.) (4.4.1.) $\forall k\in K$: $(k \sqsubseteq \top)$ und
(4.4.2) $\forall k\in K$: $(\bot \sqsubseteq k)$,

(5.) eine *Äquivalenzrelation* $\circeq \subseteq (K \times K)$
mit (5.1.) $\forall k\in K$: $(k \circeq k)$,
(5.2.) $\forall k_1,k_2\in K$: $(k_1 \circeq k_2) \leftrightarrow (k_2 \circeq k_1)$ und
(5.3.) $\forall k_1,k_2,k_3\in K$: $((k_1 \circeq k_2) \wedge (k_2 \circeq k_3)) \rightarrow (k_1 \circeq k_3)$,

(6.) eine *Inkompatibilitätsrelation* $\curlyvee \subseteq (K \times K)$
mit (6.1.) $\forall k\in K$: $\neg(k \curlyvee k)$,
(6.2.) $\forall k_1,k_2\in K$: $(k_1 \curlyvee k_2) \leftrightarrow (k_2 \curlyvee k_1)$ und
(6.3.) $\forall k_1,k_2\in K$: $(k_1 \curlyvee k_2) \rightarrow (\forall k\in K: (k \sqsubseteq k_1) \rightarrow (k \curlyvee k_2))$,

(7.) eine Menge $OPS=\{O_1,...,O_I\}$ von *Operationssymbolen* mit $i=1,...,I$ und $I\in\mathbb{N}$,

(8.) eine *Operationssymbol-Typisierungsfunktion*
$typ_{OPS_{OS}}$: $OPS \rightarrow K^* \times K$,

(9.) eine Menge $RS=\{R_1,...,R_J\}$
von *Relationssymbolen* mit $j=1,...,J$ und $J\in\mathbb{N}$,

(10.) eine *Relationssymbol-Typisierungsfunktion*
$typ_{RS_{OS}}$: $RS \rightarrow K^+$,

(11.) eine Familie $VARF_{SIG_{OS}} = (VAR_k)_{k\in K}$ *konzeptspezifischer Variablenmengen*,

(12.) eine Familie $bezf=(bez_{lan})_{lan\in\{ger,eng,fr,...\}}$
sprachspezifischer Bezeichnungsfunktionen mit
bez_{lan}: $(K \cup OPS \cup RS) \rightarrow pot_+(ALPH_{META}{}^*)$ und

(13.) eine Familie $deff=(def_{lan})_{lan\in\{ger,eng,fr...\}}$
sprachspezifischer Definitionsfunktionen mit
def_{lan}: $(K \cup OPS \cup RS \cup FORM_{SIG_{OS}}) \rightarrow ALPH_{META}{}^*$.

Bei den Elementen der Mengen K, OPS, RS und $VARF_{SIG_{OS}}$ handelt es sich um *objektsprachliche* Konstrukte. Bei den restlichen Komponenten einer ontologischen Signatur SIG_{OS} handelt es sich um *metasprachliche* Konstrukte. Hierzu gehören die Mengenfunktion MEN, das meta- und natürlichsprachliche Alphabet $ALPH_{META}$, die Subkonzeptrelation $\sqsubseteq$, die Äquivalenzrelation $\triangleq$, die Inkompatibilitätsrelation $\curlyvee$, die Familie bezf sprachspezifischer Bezeichnungsfunktionen, die Familie deff sprachspezifischer Definitionssfunktionen, die ontologische Operationssymbol-Typisierungsfunktion $typ_{OPS_{OS}}$ und die ontologische Relationssymbol-Typisierungsfunktion $typ_{RS_{OS}}$.

Zentrale Strukturierungseinheiten jeder ontologischen Signatur SIG_{OS} sind *Konzepte*. Sie werden dazu verwendet, die Operations- und Relationssymbole aus SIG_{OS} zu typisieren. Entsprechend dieser Typisierung werden zulässige Terme und Formeln über SIG_{OS} formuliert. Hinsichtlich dieser Funktionalität stimmen Konzepte mit Sorten aus der sortierten Prädikatenlogik überein.

Die meisten Konzepte werden in *einwertige* und *mengenwertige* Konzepte unterschieden. Während die Menge K_{EW} alle einwertigen Konzepte umfasst, sind in der Menge K_{MW} alle mengenwertigen Konzepte enthalten. Über ein- und mengenwertige Konzepte hinaus sind in einer ontologischen Signatur SIG_{OS} die zwei ausgezeichneten Konzepte $\top$ und $\bot$ enthalten. Auf die materiale und formale Bedeutung der ausgezeichneten Konzepte $\top$ und $\bot$ und der Unterscheidung zwischen ein- und mengenwertigen Konzepten wird im Rahmen der Untersuchung der intensionalen Semantik von Konzepten näher eingegangen.[1)]

Hinsichtlich der weiteren objektsprachlichen Konstrukte unterscheidet sich eine ontologische Signatur SIG_{OS} nicht von einer sortierten Signatur SIG_{SS}. Der vermeintliche Unterschied zwischen *Konzepten* und *Sorten* wird dadurch aufgelöst, dass die beiden zugehörigen Bezeichnungen zueinander synonym gesetzt werden.[2)]

Wesentliche Unterschiede zwischen ontologischen und sortierten Signaturen liegen hingegen in der reichhaltigen Verfügbarkeit von metasprachlichen Ausdrucksmitteln. So stehen z.B. in sortierten Signaturen die *Strukturierungsrelationen* $\sqsubseteq$, $\triangleq$ und $\curlyvee$ nicht zur Verfügung. Mit Hilfe der Strukturierungsrelationen werden in ontologischen Signaturen Beziehungen zwischen Konzepten ausgedrückt, die es bei der Grammatik zur Konstruktion von Termen über SIG_{OS} zu berücksichtigen gilt. Darüber hinaus können objektsprachlichen Ausdrücken über ontologischen Signaturen natürlichsprachliche Bezeichnungen (bezf) und Definitionen (deff) zugeordnet werden. Die Zuordnung natürlichsprachlicher Definitionen wird auch für Formeln ermöglicht, die über einer ontologischen Signatur SIG_{OS} konstruiert werden können. Sowohl Bezeichnungen als auch De-

1) Vgl. Abschnitt 3.1.3.2.1.1.

2) Allerdings bedarf es weiterführender Erläuterungen, um eine solche Synonymie zuzulassen. Auf die Synonymie-Beziehung zwischen den Bezeichnern „Konzept“ und „Sorte“ wird in Abschnitt 3.1.3.2.1.1 eingegangen.

finitionen werden als metasprachliche Zeichenketten über dem Alphabet $ALPH_{META}$ konstruiert.

Operationssymbole werden in einer ontologischen Signatur SIG_{OS} mit Hilfe der Operationssymbol-Typisierungsfunktion

$$typ_{OPS_{OS}}: OPS \rightarrow K^* \times K$$

typisiert.

Für ein Operationssymbol $O_i \in OPS$ mit $typ_{OPS_{OS}}(O_i)=(k_1...k_n,k_{n+1})$ wird die Folge $k_1...k_n \in K^*$ der Konzepte $k_1,...,k_n \in K$ als *Argumentbereich* von O_i bezeichnet. Das Konzept k_{n+1} wird entsprechend als *Zielkonzept* bezeichnet.

Analog zu dem Vorgehen bei Operationssymbolen aus sortierten Signaturen kann mittels der Funktionen

$$ARG_{OPS_{OS}}: OPS \rightarrow K^*$$
$$\text{mit } \forall O_i \in OPS: typ_{OPS_{OS}}(O_i)=(k_1...k_n,k_{n+1}) \rightarrow ARG_{OPS_{OS}}(O_i)=(k_1...k_n)$$

und

$$ZIEL_{OPS_{OS}}: OPS \rightarrow K$$
$$\text{mit } \forall O_i \in OPS: typ_{OPS_{OS}}(O_i)=(k_1...k_n,k_{n+1}) \rightarrow ZIEL_{OPS_{OS}}(O_i)=k_{n+1}$$

auf den Argument- bzw. den Zielbereich von Operationssymbolen zugegriffen werden.[1)]

Jedes null-stellige Operationssymbol O_i, das in der Form $typ_{OPS_{SS}}(O_i)=(w,k)$ mit $w=\lambda$ und $k \in K$ typisiert ist, wird als *Konstantensymbol* zum Konzept k bezeichnet. Die konzeptspezifische Menge KON_k umfasst alle Konstantensymbole zum Konzept k. Die konzeptspezifischen Mengen von Konstantensymbolen werden von der Familie $KONF_{SIG_{OS}}$ umfasst:

1) Bei der Typisierung sowohl von Operations- als auch von Relationssymbolen wird offensichtlich, warum es sich empfiehlt, als erstes Zeichen aus Konzepten jeweils Großbuchstaben zu verwenden. Dieses Vorgehen kann notwendig sein, um die Teilfolgen einer Konzeptfolge $w \in K^*$ identifizieren zu können. Beispielsweise kann im Fall der Konzeptfolge AutoBahn$\in K^*$ eindeutig bestimmt werden, dass es sich um eine Folge der Konzepte Auto,Bahn$\in K$ handelt, wenn zuvor klargestellt wurde, dass nur das erste Zeichen von Konzepten Großbuchstaben sein dürfen. Daher würde es keine Probleme bereiten, wenn ein weiteres Konzept Autobahn$\in K$ existieren würde. Wenn allerdings die Konzepte durchgehend entweder klein oder groß geschrieben werden, könnte die Konzeptfolge autobahn$\in K^*$ auf zwei unterschiedliche Weisen interpretiert werden. Bei der ersten Interpretation würde die Konzeptfolge autobahn$\in K^*$ aus dem Konzept autobahn$\in K$ hervorgehen. Dies ist zulässig, da Konzepte als Konzeptfolgen zugelassen wurden. In der zweiten Interpretation würde die Konzeptfolge autobahn$\in K^*$ aus der Konkatenation der Konzepte auto,bahn$\in K$ hervorgehen. Um solche Ambiguitäten auszuschließen, wird gefordert, dass die ersten Zeichen aller Konzepte groß und die restlichen Buchstaben klein notiert werden. Darüber hinaus wird im weiteren Verlauf bei der Notation von Konzepten grundsätzlich ein Leerschritt zwischen allen Konzepten aus einer Konzeptfolge gesetzt.

$$\mathrm{KONF}_{\mathrm{SIG}_{\mathrm{OS}}} = (\mathrm{KON}_k)_{k \in K}$$
$$\text{mit } \mathrm{KON}_k = \{O_i \mid O_i \in \mathrm{OPS} \wedge \mathrm{ARG}_{\mathrm{OPS}_{\mathrm{OS}}}(O_i) = (\lambda)\}.$$

Relationssymbole werden in einer ontologischen Signatur $\mathrm{SIG}_{\mathrm{OS}}$ mit Hilfe der Relationssymbol-Typisierungsfunktion

$$\mathrm{typ}_{\mathrm{RS}_{\mathrm{OS}}}: \mathrm{RS} \rightarrow K^+$$

auf eine Konzeptfolge $w \in K^+$ abgebildet. Der Typ $\mathrm{typ}_{\mathrm{RS}_{\mathrm{OS}}}(R_j) = (k_1 \ldots k_n)$ zu einem Relationssymbol R_j entspricht dessen *Argumentbereich*. Der Argumentbereich von Relationssymbolen kann mit jeder nicht-leeren Konzeptfolge $w \in (K^* \backslash \lambda)$ typisiert werden. Das leere Wort λ stellt den aussagenlogischen Grenzfall dar und wird in Ontologien nicht berücksichtigt.

Die *extensionale* Semantik einer ontologischen Signatur $\mathrm{SIG}_{\mathrm{OS}}$ wird mit Hilfe einer $\mathrm{SIG}_{\mathrm{OS}}$-Struktur $A_{\mathrm{SIG}_{\mathrm{OS}}}$ angegeben. Hinsichtlich ihrer extensionalen Interpretation stimmen ontologische Signaturen somit weitestgehend mit konventionellen oder sortierten Signaturen überein.

Darüber hinaus können allerdings mit den vielfachen metasprachlichen Ausdrucksmitteln aus einer ontologischen Signatur $\mathrm{SIG}_{\mathrm{OS}}$ auch Aspekte der *intensionalen* Semantik von deskriptiven Symbolen berücksichtigt werden. Dabei erfolgt die Berücksichtigung der intensionalen Semantik deskriptiver Symbole sowohl *formal* als auch *informal*.

Die *formale* Berücksichtigung der intensionalen Semantik wird neben den Typisierungsfunktionen $\mathrm{typ}_{\mathrm{OPS}_{\mathrm{OS}}}$ und $\mathrm{typ}_{\mathrm{RS}_{\mathrm{OS}}}$ zusätzlich durch die Strukturierungsrelationen $\sqsubseteq$, $\triangleq$ und $\curlyvee$ gewährleistet. Durch die Elemente der Strukturierungsrelationen werden Konzepte derart strukturiert, dass aufgrund ihrer intensionalen Semantik die Extensionen in entsprechenden Beziehungen zueinander stehen müssen. Mit Hilfe der Typisierungsfunktionen werden Operations- und Relationssymbole derart „genormt“, dass nur solche Ausdrücke als zulässig gelten, die mit den Intensionen der entsprechenden Symbole konform sind.

Die *informale* intensionale Semantik deskriptiver Symbole wird durch zwei Komponenten ontologischer Signaturen angegeben. Die erste Komponente ist die Familie bezf sprachspezifischer Bezeichnungsfunktionen. Mit Hilfe jeder sprachspezifischen Bezeichnungsfunktion können sowohl Konzepten als auch Operations- und Relationssymbolen solche metasprachlichen Zeichenketten zugeordnet werden, die als natürlichsprachliche *Bezeichner* formalsprachlicher Konstrukte dienen. Die zweite Komponente deff umfasst sprachspezifische Definitionsfunktionen. Mit Hilfe der Definitionsfunktionen können neben den Konzepten auch Operations- und Relationssymbolen metasprachliche Zeichenketten als natürlichsprachliche *Definitionen* zugeordnet werden. Zudem können sprachspezifische Definitionsfunktionen in ihren Argumenten auch Formeln über ontologischen Signaturen aufnehmen. Diese Möglichkeit erweist sich dann als besonders nützlich, wenn komplexe Formeln über einer ontologischen Signatur als *Regeln* in einer ontologischen Spezifikation verwendet werden sollen und ihre Verarbeitbarkeit menschlichen Akteuren mit unter Umständen geringen Erfahrungen im Umgang mit formalen Sprachen erleichtert werden soll.

3.1.2.2 Ontologische Ausdrücke

3.1.2.2.1 Ontologische Terme

Analog zu den Mengen $EXPR_{SIG_{KS}}$ und $EXPR_{SIG_{SS}}$ aller konventionellen bzw. sortierten Ausdrücke umfasst die Menge

$$EXPR_{SIG_{OS}} \subset ALPH_{OS}^*$$

alle Ausdrücke über einer ontologischen Signatur SIG_{OS}. Ausdrücke über einer ontologischen Signatur SIG_{OS} werden im Folgenden auch als ontologische Ausdrücke bezeichnet, wenn aus dem Argumentationskontext nicht hervorgeht, über welchem Signaturtyp (konventionell, sortiert oder ontologisch) die Aussage konstruiert ist.

Die Menge $EXPR_{SIG_{OS}}$ aller Ausdrücke über einer ontologischen Signatur SIG_{OS} umfasst zum einen die Menge $TERM_{SIG_{OS}}$ aller *ontologischen Terme* und zum anderen die Menge $FORM_{SIG_{OS}}$ aller *ontologischen Formeln* über SIG_{OS}.

$$EXPR_{SIG_{OS}} = TERM_{SIG_{OS}} \cup FORM_{SIG_{OS}}$$
$$\text{mit } TERM_{SIG_{OS}} \cap FORM_{SIG_{OS}} = \varnothing$$

Analog zu der Definition der Menge $FORM_{SIG_{KS}}$ aller konventionellen Formeln und der Definition der Menge $FORM_{SIG_{SS}}$ aller sortierten Formeln werden zur Definition ontologischer Formeln Terme über einer ontologischen Signatur SIG_{OS} vorausgesetzt. Daher werden im Folgenden zunächst ontologische Terme vorgestellt.

Jedem ontologischen Term $t \in TERM_{SIG_{OS}}$ ist ein Konzept $k \in K$ als Typ zugewiesen. Ontologische Terme, die ein Konzept $k \in K$ als Typ haben, werden zu einer konzeptspezifischen Termmenge $TERM_k$ zusammengefasst. Jede konzeptspezifische Termmenge $TERM_k$ umfasst demnach Terme zu einem Konzept $k \in K$. Die Vereinigung aller konzeptspezifischen Termmengen ist die Menge $TERM_{SIG_{OS}}$ aller Terme über einer ontologischen Signatur SIG_{OS}:

$$TERM_{SIG_{OS}} = \cup_{k \in K} TERM_k$$

Konzeptspezifische Termmengen werden zur Familie $TERMF_{SIG_{OS}}$ konzeptspezifischer Termmengen über einer ontologischen Signatur SIG_{OS} zusammengefasst. Bei den Mitgliedern der Mengenfamilie $TERMF_{SIG_{OS}}$ handelt es sich entsprechend um konzeptspezifische Termmengen:

$$TERMF_{SIG_{OS}} = (TERM_k)_{k \in K}.$$

Die Definition von ontologischen Termen ist grundsätzlich der Definition von sortierten Termen ähnlich. Ontologische Terme können einerseits aus Variabeln und andererseits aus Operationssymbolen und deren typgerechter Anwendung auf ontologische Terme hervorgehen. Im Gegensatz zu konventionellen Termen ist sowohl bei ontologischen als auch bei sortierten Termen der Typ des Operationssymbols nicht nur *numerisch* bestimmt. Der Typ $typ_{OPS_{OS}}(O_i)$ jedes Operationssymbols ist immer ein Zwei-Tupel (w,k) bestehend aus einer Konzeptfolge $w \in K^*$ und einem Konzept $k \in K$. Der Typ $typ_{OPS_{OS}}(O_i)$ jedes Operationssymbols O_i ist somit sowohl numerischer Art, nämlich hinsichtlich der

Länge der Konzeptfolge w, als auch *inhaltlicher* Art, und zwar in Bezug auf seine Zusammensetzung aus Konzepten aus K.[1)] Hinsichtlich der Typisierungen von Operationssymbolen, die bei der Konstruktion von Termen zu beachten sind, unterscheiden sich somit Ontologien nicht von der sortierten Prädikatenlogik.

Darüber hinaus sind jedoch bei der Definition von ontologischen Termen *Erweiterungen* gegenüber der Definition von sortierten Termen zu beachten. Die erste Erweiterung betrifft die Termdefinition aufgrund der Subkonzeptrelation $\sqsubseteq$. Die zweite Erweiterung betrifft die Termdefinition aufgrund der *Äquivalenzrelation* $\triangleq$. Die dritte Erweiterung betrifft den Ausschluss von Termen aufgrund der Inkompatibilitätsrelation $\curlyvee$.

Eine konzeptspezifische Menge $TERM_k$ von Termen zu einem Konzept $k \in K$ sind wie folgt definiert:

(1.) Wenn $x_q \in VAR_k$ gilt, dann gilt auch $x_q \in TERM_k$. Jede Variable aus einer konzeptspezifischen Variablenmenge VAR_k ist ein *atomarer ontologischer Term* zum Konzept k:

$$\forall k \in K: VAR_k \subseteq TERM_k.$$

(2.) Wenn $O_i \in OPS$ mit $typ_{OPS_{OS}}(O_i) = (\lambda, k)$, $\lambda \in K^*$ und $k \in K$ gilt, dann gilt auch $O_i \in TERM_k$. Jedes Konstantensymbol $O_i \in KON_k$ ist ein *atomarer ontologischer Term* zum Konzept k:

$$\forall O_i \in OPS: typ_{OPS_{OS}}(O_i) = (\lambda, k) \rightarrow O_i \in TERM_k.$$

(3.) Wenn $O_i \in OPS$ mit $typ_{OPS_{OS}}(O_i) = (k_1 ... k_n, k)$, $k_1 ... k_n \in K^*$ und $k \in K$ gilt und wenn $t_1, ..., t_n$ jeweils ontologische Terme mit $t_x \in TERM_{k_x}$ mit $x = 1, ..., n$ und $n \in \mathbb{N}_+$ sind, dann ist $O_i(t_1, ..., t_n)$ ein *zusammengesetzter ontologischer Term* zum Konzept k:

$$\forall O_i \in OPS_{OS}, \exists n \in \mathbb{N}_+, \exists k_1, ..., k_n, k \in K:$$
$$typ_{OPS_{OS}}(O_i) = (k_1 ... k_n, k) \wedge (\forall x \in \{1, ..., n\}: t_x \in TERM_{k_x}) \rightarrow$$
$$O_i(t_1, ... t_n) \in TERM_k.$$

(4.) Wenn $(k_1 \sqsubseteq k_2)$ gilt, ist jeder ontologische Term zum Konzept $k_1 \in K$ auch ein ontologischer Term zum Konzept $k_2 \in K$:

$$\forall k_1, k_2 \in K: (k_1 \sqsubseteq k_2) \rightarrow (TERM_{k_1} \subseteq TERM_{k_2})$$

(5.) Wenn $(k_1 \triangleq k_2)$ gilt, ist jeder ontologische Term zum Konzept $k_1 \in K$ auch ein ontologischer Term zum Konzept $k_2 \in K$ und umgekehert:

$$\forall k_1, k_2 \in K: (k_1 \triangleq k_2) \rightarrow (TERM_{k_1} = TERM_{k_2}).$$

(6.) Wenn $(k_1 \curlyvee k_2)$ gilt, ist kein ontologischer Term $t \in TERM_{k_1}$ zu einem Konzept $k_1 \in K$ auch ein ontologischer Term zum Konzept $k_2 \in K$:

$$\forall k_1, k_2 \in K: (k_1 \curlyvee k_2) \rightarrow (TERM_{k_1} \cap TERM_{k_2} = \varnothing).$$

Wie bei der Definition von konventionellen und sortierten Termen, so finden Relationssymbole bei der Definition von ontologischen Termen auch keine Berücksichtigung.

1) Analog sind auch die Typen von Relationssymbolen in ontologischen Signaturen nicht nur numerisch, sondern auch inhaltlich charakterisiert. Allerdings sind Relationssymbole für die Konstruktion von Termen über ontologischen Signaturen nicht von Bedeutung.

Ontologische Terme gehen entweder aus konzeptspezifischen Variablen, aus Konstantensymbolen oder aus der typgerechten Anwendung von Operationssymbolen auf Terme hervor.

Die Regeln (1.), (2.) und (3.) sind analoge Übertragungen der Regeln (1.), (2.) und (3.) aus Abschnitt 2.2.1.2.2.2.1 zur Definition von sortierten Termen. Mit den Regeln (1.), (2.) und (3.) werden die Strukturierungsprinzipien bei der Definition von Termen über einer sortierten Signatur SIG_{SS} auf die Definition von Termen über einer ontologischen Signatur SIG_{OS} übertragen.

Ontologische Terme, die aus der Anwendung von Regel (1.) oder Regel von (2.) hervorgehen, werden auch als *atomare* ontologische Terme bezeichnet.[1)] Atomare ontologische Terme gehen demnach entweder aus Variablen oder aus Konstantensymbolen hervor. Ontologische Terme, die aus der Anwendung von Regel (3.) hervorgehen, werden als *zusammengesetzte* oder *komplexe* ontologische Terme bezeichnet. Zusammengesetzte ontologische Terme gehen demnach aus der typgerechten Anwendung von ontologischen Operationssymbolen auf andere ontologische Terme hervor.

Wesentliche Erweiterungen gegenüber den Definitionen konventioneller und sortierter Terme sind die Regeln (4.) bis (7.). Mit Hilfe der Strukturierungsrelationen $\sqsubseteq$, $\mathrel{\mathring{=}}$ und $\curlyvee$ wird der Familie $TERMF_{SIG_{OS}}$ aller konzeptspezifischen Termmengen über einer ontologischen Signatur SIG_{OS} eine *innere Struktur* aufgesetzt, die über die Strukturierung der Familie $TERMF_{SIG_{SS}}$ hinausgeht. Für die einzelnen konzeptspezifischen Termmengen werden durch die Regeln (4.) bis (7.) mengentheoretische Beziehungen, die zwischen den Termmengen zu gelten haben, formuliert. Eine solche innere Struktur ist weder für die Familie $TERMF_{SIG_{SS}}$ aller Termmengen über einer sortierten Signatur SIG_{SS} noch für die Menge $TERM_{SIG_{KS}}$ aller Terme über einer konventionellen Signatur SIG_{KS} vorgesehen.

Während die Menge $TERM_{SIG_{KS}}$ grundsätzlich keine Differenzierung ihrer Elemente erlaubt, sind die Mitglieder der Familie $TERMF_{SS}$ aller sortenspezifischen Termmengen vollkommen unabhängig voneinander definiert. Für sortierte Terme kann lediglich ihre Zugehörigkeit zu einer sortenspezifischen Termmenge aufgrund der Typisierungen in der sortierten Signatur SIG_{SS} ausgedrückt werden. Atomare sortierte Terme gehören zu einer sortenspezifischen Termmenge $TERM_{s_1}$, wenn die Variablen oder Konstantensymbole, aus denen sie hervorgehen, mit der Sorte $s_1 \in S$ typisiert sind. Zusammengesetzte sortierte Terme gehören hingegen zu einer sortenspezifischen Termmenge $TERM_{s_1}$, wenn das Operationssymbol O_i, aus dessen Anwendung auf andere sortierte Terme sie hervorgehen, im Zielbereich $ZIEL_{OPS_{SS}}(O_i)$ mit der Sorte $s_1 \in S$ typisiert ist. Darüber hinaus kann die Zugehörigkeit eines sortierten Terms zu einer zweiten sortenspezifischen Termmenge $TERM_{s_2}$ mit $s_2 \in S$ mit keinen anderen metasprachlichen Ausdrucksmitteln eingefordert werden. Mit den Regeln (4.), (5.) und (6.) wird diese Spezi-

1) Der Zusatz „ontologische" wird im Folgenden ausgelassen, wenn aus dem Argumentationskontext hervorgeht, dass es sich um einen Term über einer ontologischen Signatur SIG_{OS} handelt.

fikationsmöglichkeit für ontologische Terme erschlossen. Durch diese Regeln können entweder – im Fall der Strukturierungsrelationen $\sqsubseteq$ und $\stackrel{\circ}{=}$ – die Zugehörigkeiten von ontologischen Termen zu unterschiedlichen konzeptspezifischen Termmengen oder – im Fall der Strukturierungsrelation $\curlyvee$ – der Ausschluss von ontologischen Termen aus konzeptspezifischen Termmengen spezifiziert werden.

Um Terme, die aufgrund einer der Regeln (1.) bis (3.) zu einer konzeptspezifischen Termmenge $TERM_{k_1}$ zugeordnet werden, von Termen unterscheiden zu können, die aufgrund der Regel (5.) oder der (6.) zur Termmenge $TERM_{k_1}$ gezählt werden, werden Erstgenannte als *originär* dem Konzept k_1 zugeordnete Terme bezeichnet. *Derivativ* dem Konzept k_1 zugeordnete Terme sind hingegen solche Terme, die originär einem anderen Konzept k_2 zugeordnet sind und aufgrund ($k_2 \sqsubseteq k_1$) oder ($k_2 \stackrel{\circ}{=} k_1$) auch der konzeptspezifischen Termmenge $TERM_{k_1}$ zugeordnet werden.[1)]

Durch die Regel (4.) der Definition von ontologischen Termen wird sichergestellt, dass alle ontologischen Terme, die einem Konzept $k_1 \in K$ zugeordnet sind, auch zu den Konzepten $k_2,...,k_n \in K$ gehören, mit denen k_1 in Subkonzeptrelation ($k_1 \sqsubseteq k_i$) mit $i \in \{2,...,n\}$ steht.[2)] Dabei können die konzeptspezifischen Termmengen $TERM_{k_2},...,TERM_{k_n}$ jeweils auch solche ontologischen Terme umfassen, die nicht in $TERM_{k_1}$ enthalten sind. Somit ist die konzeptspezifische Termmenge $TERM_{k_1}$ stets eine (unechte)[3)] Teilmenge einer konzeptspezifischen Termmenge $TERM_{k_2}$, wenn ($k_1 \sqsubseteq k_2$) gilt:

$$\forall k_1,k_2 \in K: (k_1 \sqsubseteq k_2) \rightarrow (TERM_{k_1} \subseteq TERM_{k_2}).$$

Diese Erweiterung wird sich später bei der Definition von Formeln über einer ontologischen Signatur SIG_{OS} darin auswirken, dass die Menge der ontologischen Terme, die im Argument von Relationssymbolen vorkommen dürfen, um die Vereinigung der konzeptspezifischen Termmengen erweitert wird, die den Subkonzepttermen aufgrund der oben angegebenen Regel zugeordnet sind.

---

1) Die Eigenart eines ontologischen Terms $t \in TERM_{k_1}$, dem Konzept $k_1 \in K$ entweder originär oder derivativ zugeordnet zu sein, ist somit eine „relationale“ Eigenschaft, die ontologische Terme mit Konzepten verknüpft.

2) Vgl. KANEIWA (2001), S. 28.

3) Eine konzeptspezifische Termmenge $TERM_{k_1}$ ist genau dann eine *echte* Teilmenge einer konzeptspezifischen Termmenge $TERM_{k_2}$, wenn einerseits alle ontologischen Terme $t_i \in TERM_{k_1}$ auch in $TERM_{k_2}$ enthalten sind und andererseits in $TERM_{k_2}$ mindestens ein ontologischer Term $t_j \in TERM_{k_2}$ enthalten ist, der in $TERM_{k_1}$ nicht enthalten ist. Die echte Teilmengenrelation wird in der Form

$$TERM_{k_1} \subset TERM_{k_2}$$

angegeben. Die konzeptspezifische Termmenge $TERM_{k_1}$ ist genau dann eine *unechte* Teilmenge der konzeptspezifischen Termmenge $TERM_{k_2}$, wenn auch $TERM_{k_1}=TERM_{k_2}$ gelten kann und somit nicht notwendig in $TERM_{k_2}$ ontologische Terme enthalten sein müssen, die nicht in $TERM_{k_1}$ enthalten sind. Die unechte Teilmengenrelation wird in der Form

$$TERM_{k_1} \subseteq TERM_{k_2}$$

angegeben.

Alle Konzepte $k_n \in K$ aus einer ontologischen Signatur SIG_{OS} stehen in Subkonzeptrelation $\sqsubseteq$ zum Maximalkonzept $\top$. Somit ist jede konzeptspezifische Termmenge $TERM_k$ zu einem beliebigen Konzept $k \in K$ eine Teilmenge der Menge $TERM_\top$. Entsprechend stimmt die Menge $TERM_{SIG_{OS}}$ aller ontologischen Terme mit der Menge $TERM_\top$ überein:

$$TERM_{SIG_{OS}} = TERM_\top.$$

Daher können in den Argumentstellen von Relations- und Operationssymbolen, die mit $\top \in K$ typisiert sind, alle ontologischen Terme eingesetzt werden.

In der konzeptspezifischen Termmenge $TERM_\perp$ zu Minimalkonzept $\top$ sind keine ontologischen Terme enthalten. Somit stimmt die konzeptspezifische Termmenge $TERM_\perp$ zum Minimalkonzept $\perp$ mit der leeren Menge $\varnothing$ überein:

$$TERM_\perp = \varnothing.$$

Die Subkonzeptrelation $\sqsubseteq$ ist in der Definition ontologischer Signaturen als Ordnungsrelation mit den Eigenschaften der Reflexivität, Antisymmetrie und Transitivität definiert.[1] Aufgrund der Transitivität der Subkonzeptrelation können Termmengen ineinander verschachtelt sein. Es gilt beispielsweise

$$\forall k_1,k_2,k_3 \in K:$$
$$(k_1 \sqsubseteq k_2 \sqsubseteq k_3) \rightarrow (TERM_{k_1} \subseteq TERM_{k_2} \subseteq TERM_{k_3})$$

Entsprechend können im Argument von Operationssymbolen, die in der Form $typ_{OPS_{OS}}(O_i)=(k_1...k_n,k_{n+1})$ typisiert sind, an den jeweiligen Argumentstellen auch solche Terme verwendet werden, die nicht originär den Termmengen $TERM_{k_1},...,TERM_{k_n}$ zugeordnet sind, sondern aufgrund der Regel (4.) zur Bestimmung konzeptspezifischer Termmengen der jeweiligen konzeptspezifischen Termmenge derivativ zugeordnet sind.

Ebenso wird der Term, der aus der Anwendung eines Operationssymbols O_i mit $typ_{OPS_{OS}}(O_i)=(k_1...k_n,k_{n+1})$ auf ein Termtupel $(t_1,...,t_n)$ hervorgeht, nicht nur der Termmenge $TERM_{k_{n+1}}$ zugerechnet, sondern auch allen Termmengen $TERM_{k_{n+1.i}}$, zu denen die Termmenge $TERM_{k_{n+1}}$ aufgrund der Subkonzeptbeziehung $(k_{n+1} \sqsubseteq k_{n+1.i})$ in Teilmengenbeziehung steht. Der Zusammenhang ist in Abbildung 3 dargestellt.

1) Vgl. Punkt (4.) in Abschnitt 3.1.2.1.

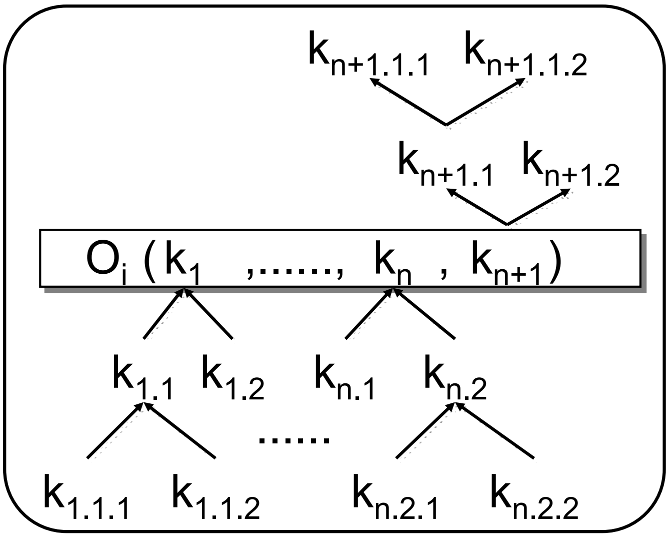

Abbildung 3: Termmengenverschachtelung und Operationssymboltypisierung

In der Abbildung 3 ist ein Operationssymbol $O_i \in OPS$ mit der Typisierung $typ_{OPS_{OS}}(O_i)=(k_1...k_n,k_{n+1})$ illustriert. Die Pfeile repräsentieren jeweils Elemente der Subkonzeptrelation $\sqsubseteq$. An der ersten Argumentstelle des Operationssymbols O_i dürfen beispielsweise – entsprechend seiner Typisierung – nur Terme zum Konzept k_1 eingesetzt werden. Dabei umfasst die konzeptspezifische Termmenge $TERM_{k_1}$ alle Elemente der Termmengen $TERM_{k_{1.i}}$, wenn die jeweiligen Konzepte in Subkonzeptrelation ($k_{1.i} \sqsubseteq k_1$) stehen. Für die weiteren Argumentstellen wird analog verfahren.[1)]

Die zweite Erweiterung durch Regel (5.) ergibt sich aus der *Äquivalenzrelation* $\triangleq$. Demnach sind alle Terme, die zu einem Konzept $k_1 \in K$ gültig sind, auch für alle Konzepte $k_2,...,k_n \in K$ gültig, mit denen k_1 in Äquivalenzrelation ($k_1 \triangleq k_i$) mit $i \in \{2,...,n\}$ steht. Da de Äquivalenzrelation $\triangleq$ als *symmetrische* Relation ausgezeichnet ist, gilt stets die Gleichheit von Termmengen zu äquivalenten Konzepten:

$$\forall k_1,k_2 \in K: (k_1 \triangleq k_2) \rightarrow (TERM_{k_1} = TERM_{k_2}).$$

Die konzeptspezifischen Termmengen $TERM_{k_1}$ und $TERM_{k_2}$ können demnach keine Terme enthalten, die in der jeweils anderen Termmenge nicht enthalten sind, wenn die beiden Konzepte k_1 und k_2 in Äquivalenzrelation $\triangleq$ miteinander stehen.

An den entsprechenden Argumentstellen eines Operationssymbols $O_i \in OPS$ mit $typ_{OPS_{OS}}(O_i)=(k_1...k_n,k_{n+1})$ können demnach jeweils auch Terme aus den Termmengen zu solchen Konzepten verwendet werden, die mit den Konzepten im Argumentbereich von O_i in Äquivalenzrelation $\triangleq$ stehen. Ebenso ist ein Term t, der aus der typgerechten

1) Ein bemerkenswerter Sonderfall tritt dann ein, wenn ein Konzept k_i in Subkonzeptrelation zu zwei verschiedenen Konzepten k_1 und k_2 steht, ohne dass k_1 und k_2 in Subkonzeptrelation zueinander stehen müssen. Ein solcher Fall wird als *Kreuzklassifikation* oder *Verzweigung der Subkonzeptrelation* bezeichnet und wird später näher untersucht; vgl. Abschnitt 3.1.3.2.2.1. In einem solchen Fall kann der gleiche Term in zwei verschiedenen Stellen verwendet werden, obwohl die Stellen mit unterschiedlichen Konzepten typisiert sind.

Anwendung des Operationssymbols O_i auf die Terme $t_1,...,t_n$ hervorgeht, nicht nur in der Termmenge $TERM_{k_{n+1}}$ enthalten, sondern auch in allen Termmengen $TERM_{k_{n+1.i}}$ mit $(k_{n+1.i} \triangleq k_{n+1})$.

Während mit den Regeln (1.) bis (5.) eine *Erweiterung* der konzeptspezifischen Termmenge $TERM_k$ verbunden ist, werden mit der Regel (6.) Terme zu k *ausgeschlossen*. Sie führt dazu, dass zwei konzeptspezifische Termmengen $TERM_{k_1}$ und $TERM_{k_2}$ disjunkt zueinander sein müssen, wenn sie mit solchen Konzepten $k_1,k_2 \in K$ typisiert sind, die miteinander in Inkompatibilitätsrelation $\curlyvee$ stehen:

$$\forall k_1,k_2 \in K: (k_2 \curlyvee k_1) \rightarrow (TERM_{k_1} \cap TERM_{k_2} = \varnothing).$$

Zur formalen Bestimmung der originären Typen ontologischer Terme wird die Termtypisierungsfunktion

$$typ_{T_{OS}}: TERM_{SIG_{OS}} \rightarrow K$$

verwendet. Sie ordnet jedem ontologischen Term $t \in TERM_{SIG_{OS}}$ ein Konzept $k \in K$ als dessen originären Typ zu. Die Bilder der Funktion $typ_{T_{OS}}$ sind wie folgt definiert:

(1.) $\forall t \in TERM_{SIG_{OS}}$: $t \in VAR_k \rightarrow typ_{T_{OS}}(t)=k$,

(2.) $\forall t \in TERM_{SIG_{OS}}$: $t \in KON_k \rightarrow typ_{T_{OS}}(t)=k$ und

(3.) $\forall t \in TERM_{SIG_{OS}}$: $(t=O_i(t_1,...,t_n) \wedge O_i \in OPS \wedge typ_{OPS_{OS}}(O_i)=(k_1...k_n,k_{n+1})$
$\wedge (\forall x=1,...,n: t_x \in TERM_{k_x})) \rightarrow typ_{T_{OS}}(t)=ZIEL_{OPS_{OS}}(O_i)=k_{n+1}$.

Der originäre Typ $typ_{T_{OS}}(t)$ eines ontologischen Terms $t \in TERM_{SIG_{OS}}$, der eine Variable ist, stimmt mit dem Konzept $k \in K$ überein, dem die entsprechende Variable zugeordnet ist. Ebenso stimmt der originäre Typ $typ_{T_{OS}}(t)$ eines ontologischen Terms $t \in TERM_{SIG_{OS}}$, der aus einem Konstantensymbol hervorgegangen ist, mit dem Konzept $k \in K$ überein, dem das Konstantensymbol zugeordnet ist. Schließlich stimmt der originäre Typ $typ_{T_{OS}}(t)$ eines ontologischen Terms $t \in TERM_{SIG_{OS}}$, der aus der typgerechten Anwendung eines Operationssymbols $O_i \in OPS$ auf n Terme hervorgegangen ist, mit dem Zielkonzept $k_{n+1} \in K$ von O_i überein.

Bei der Zuweisung eines Konzepts $k \in K$ als originärer Typ eines ontologischen Terms $t \in TERM_{SIG_{OS}}$ werden demnach lediglich die Konstruktionsregeln berücksichtigt, die in analoger Weise für die konventionelle und sortierte Prädikatenlogik auch gelten. Die Zuordnung eines ontologischen Terms aus einer konzeptspezifischen Termmenge $TERM_k$ zu allen konzeptspezifischen Termmengen $TERM_{k_n}$, für die $(k \sqsubseteq k_n)$ oder $(k \triangleq k_n)$ gilt, wirkt sich nicht auf originäre die Typzuordnung aus. Entsprechend den neu hinzugekommenen Regeln der Termdefinition kann eine konzeptspezifische Termmenge $TERM_k$ auch solche ontologischen Terme $t_x \in TERM_{SIG_{OS}}$ umfassen, für die nicht $typ_{T_{OS}}(t_x)=k$ gilt. Dies ist genau dann der Fall, wenn die ontologischen Terme t_x mit $t_x \in TERM_{k_n}$ mit $k_n \neq k$ und $k_n \sqsubseteq k$ bzw. $k_n \triangleq k$ aufgrund der Regel (4.) oder (5.) zur Konstruktion ontologischer Terme zu $TERM_k$ gezählt werden. Es kann allerdings festgehalten werden, dass der Typ $typ_{T_{OS}}(t)=k$ eines ontologischen Terms $t \in TERM_{k_n}$ entweder das Konzept $k_n \in K$ sein muss oder in Subkonzept- oder Äquivalenzrelation zum Konzept k_n stehen muss:

$\forall t \in TERM_{SIG_{OS}}$: $(t \in TERM_{k_n} \wedge typ_{T_{OS}}(t)=k) \rightarrow ((k = k_n) \vee (k \sqsubseteq k_n) \vee (k \mathrel{\mathring{=}} k_n))$.[1)]

Die Klassifikation von Termen über konventionellen und sortierten Signaturen kann analog auch für Terme über ontologischen Signaturen durchgeführt werden. Terme über ontologischen Signaturen können demnach auch hinsichtlich ihres *Variablenanteils* und ihrer *Zusammengesetztheit* klassifiziert werden. Darüber hinaus ist für ontologische Terme eine weitere Klassifikationsmöglichkeit vorgesehen. Diese dritte Klassifikationsmöglichkeit betrifft die *Wertigkeit* von ontologischen Termen.

Hinsichtlich ihres Variablenanteils können Terme über einer ontologischen Signatur SIG_{OS} – je nach ihrem Typ – zu den Mitgliedern der Mengenfamilien $VTF_{SIG_{OS}}$ und $GTF_{SIG_{OS}}$ gerechnet werden. Die Familie $VTF_{SIG_{OS}}$ umfasst konzeptspezifische Mengen VT_k *variabler Terme* über einer ontologischen Signatur SIG_{OS}:

$$VTF_{SIG_{OS}} = (VT_k)_{k \in K}.$$

Die Familie $GTF_{SIG_{OS}}$ umfasst konzeptspezifische Mengen GT_k von *Grundtermen* über einer ontologischen Signatur SIG_{OS}:

$$GTF_{SIG_{OS}} = (GT_k)_{k \in K}.$$

Jede konzeptspezifische Menge $TERM_k$ von ontologischen Termen setzt sich aus einer konzeptspezifischen Menge VT_k variabler ontologischer Terme und einer konzeptspezifischen Menge GT_k von ontologischen Grundtermen zusammen:

$$TERM_k = VT_k \cup GT_k \text{ für alle } k \in K.$$

Die Bestimmung des Variablenanteils von ontologischen Termen erfolgt durch die Funktion

$$var_{T_{OS}}: TERM_{SIG_{OS}} \rightarrow pot(VAR).$$

Die Funktion $var_{T_{OS}}$ ist analog zu den Funktionen $var_{T_{KS}}$ und $var_{T_{SS}}$ definiert. Demnach sind die Bilder für die Variablenfunktion $var_{T_{OS}}$ wie folgt definiert:

(1.) $var_{T_{OS}}(x_q) = \{x_q\}$
für $x_q \in VAR$,

(2.) $var_{T_{OS}}(O_i) = \varnothing$
für $O_i \in OPS$ mit $ARG_{OPS_{OS}}(O_i)=\lambda$ und

(3.) $var_{T_{OS}}(O_i(t_1,...,t_n)) = var_{T_{OS}}(t_1) \cup ... \cup var_{T_{OS}}(t_n)$
für $O_i \in OPS$ mit $typ_{OPS_{OS}}(O_i)=(k_1...k_n,k_{n+1})$.

Mit Hilfe der Funktion $var_{T_{OS}}$ können die konzeptspezifischen Mengen VT_k und GT_k variabler Terme bzw. Grundterme festgelegt werden. Die konzeptspezifische Menge

$$VT_k = \{t \mid t \in TERM_k \wedge var_{T_{OS}}(t) \neq \varnothing\}$$

---

1) Hierbei ist zu beachten, dass bei Gleichheit von k_m und k_n die Subkonzept- und die Äquivalenzrelation ebenso erfüllt sind wegen der Reflexivität der beiden metasprachlichen Relationen.

umfasst alle variablen Terme zu einem Konzept $k \in K$. Jede konzeptspezifische Menge VAR_k von Variablen ist immer eine Teilmenge der jeweiligen konzeptspezifischen Menge variabler Terme:

$$VAR_k \subseteq VT_k.$$

Jede konzeptspezifische Menge

$$GT_k = \{t \mid t \in TERM_k \wedge var_{T_{OS}}(t) = \varnothing\}$$

umfasst Grundterme zum Konzept k. Jede konzeptspezifische Menge KON_k von *Konstantensymbolen* zu einem Konzept k ist stets eine Teilmenge der entsprechenden Menge von Grundtermen:

$$KON_k \subseteq GT_k.$$

Hinsichtlich ihrer Zusammengesetztheit können Terme über einer ontologischen Signatur SIG_{OS} – je nach ihrem Typ – den konzeptspezifischen Mitgliedern der Mengenfamilien $ATF_{SIG_{OS}}$ und $ZTF_{SIG_{OS}}$ zugeordnet werden. Die Mengenfamilie $ATF_{SIG_{OS}}$ umfasst konzeptspezifische Mengen atomarer Terme:

$$ATF_{SIG_{OS}} = (AT_k)_{k \in K}$$
$$\text{mit } AT_k = \{t \mid t \in IND_{k_n} \wedge ((k_n \sqsubseteq k) \vee (k_n \triangleq k))\}.$$

Die Mengenfamilie $ZTF_{SIG_{OS}}$ umfasst konzeptspezifische Mengen zusammengesetzter Terme:

$$ZTF_{SIG_{OS}} = (ZT_k)_{k \in K}$$
$$\text{mit } ZT_k = \{t \mid t \in TERM_{k_n} \wedge (t \notin (IND_{k_n}) \wedge ((k_n \sqsubseteq k) \vee (k_n \triangleq k))\}.$$

Die Definitionen der Mengen ZT_k und AT_k zusammengesetzter bzw. atomarer Terme zu einem Konzept k unterscheiden sich von den Definitionen der entsprechenden Mengen ZT_s bzw. AT_s zu einer Sorte s. Im Fall der sortierten Prädikatenlogik reichte es aus zu bestimmen, dass ein Term ein Individuensymbol zu einer Sorte s sein muss, um zu der sortenspezifischen Termmenge $TERM_s$ gezählt zu werden. Im Fall ontologischer Terme kann allerdings die Menge AT_k auch solche Terme umfassen, die nicht der Menge IND_k von Individuensymbolen zum Konzept k zugeordnet sind. Es reicht aus, wenn es sich bei ihnen um Individuensymbole zu einem Konzept k_n handelt, das in Subkonzept- oder Äquivalenzrelation zu k steht. Insofern werden Individuensymbole, die originär einem Konzept k_n und derivativ dem Konzept k zugezählt werden, auch zu der Menge AT_k aller atomaren Terme zum Konzept k gezählt. Analog gilt, dass bei der Bestimmung der Menge ZT_k aller zusammengesetzten Terme nicht nur die originär zum k zugehörigen zusammengesetzten Terme, sondern auch alle derivativ zugeordneten zusammengesetzten Terme hierzu gezählt werden.

Neben der Unterteilung der Menge $TERMF_{SIG_{OS}}$ hinsichtlich der zuvor genannten Kriterien, die bereits aus der Unterteilung der Mengenfamilie $TERMF_{SIG_{SS}}$ bekannt sind, können Terme über ontologischen Signaturen noch hinsichtlich ihrer *Wertigkeit* unterteilt werden. Die Wertigkeit eines ontologischen Terms ist unmittelbar von der Wertigkeit des jeweiligen Konzepts ableitbar, dem der Term zugeordnet wird. Bereits in der

Definition ontologischer Signaturen wurde darauf hingewiesen, dass die Menge K aller Konzepte u.a.[1] eine Menge K_{EW} und eine Menge K_{MW} umfasst. Die Menge K_{EW} umfasst *einwertige* Konzepte. Die Menge K_{MW} umfasst hingegen *mengenwertige* Konzepte. Entsprechend der Differenzierung von Konzepten können auch Terme über ontologischen Signaturen hinsichtlich ihrer Wertigkeit unterschieden werden.

Die Mengenfamilie $TERMF_{SIG_{OS}}$ kann hinsichtlich der Wertigkeit der zugrunde gelegten ontologischen Terme wie folgt unterteilt werden:

$$\begin{gathered} TERMF_{SIG_{OS}} = ETF_{SIG_{OS}} \cup MTF_{SIG_{OS}} \\ \text{mit } ETF_{SIG_{OS}}=(ET_k)_{k \in K_{EW}}, \\ ET_k=\{t \mid t \in TERM_k \wedge k \in K_{EW}\} \\ \text{und } MTF_{SIG_{OS}}=(MT_k)_{k \in K_{MW}} \\ MT_k=\{t \mid t \in TERM_k \wedge k \in K_{MW}\} \end{gathered}$$

Die Mengenfamilie $ETF_{SIG_{OS}}=(ET_k)_{k \in K_{EW}}$ umfasst konzeptspezifische Termmengen zu einwertigen Konzepten. Entsprechend werden die Elemente jeder konzeptspezifischen Termmenge ET_k als *einwertige ontologische Terme* oder kurz einwertige Terme bezeichnet. Die Mengenfamilie $MTF_{OS}=(MT_k)_{k \in K_{MW}}$ umfasst hingegen konzeptspezifische Termmengen zu mengenwertigen Konzepten. Entsprechend werden die Elemente jeder konzeptspezifischen Termmenge MT_k als *mengenwertige ontologische Terme* oder kurz mengenwertige Terme bezeichnet.

3.1.2.2.2 Ontologische Formeln

Die zweite Teilmenge der Menge $EXPR_{SIG_{OS}}$ aller Ausdrücke über einer ontologischen Signatur SIG_{OS} umfasst Formeln über SIG_{OS}. Analog zu den Bezeichnungen konventioneller und sortierter Formeln werden Formeln über einer ontologischen Signatur SIG_{OS} im Folgenden auch als *ontologische Formeln* bezeichnet, wenn aus dem Argumentationskontext nicht ersichtlich ist, um welche Art von Formeln es sich handelt.

Die Menge $FORM_{SIG_{OS}}$ aller Formeln über einer ontologischen Signatur SIG_{OS} ist induktiv definiert als:

(1.) Es gilt $w,f \in FORM_{SIG_{OS}}$. w ist die immer gültige, tautologische Formel; f ist die immer ungültige, kontradiktorische Formel.

(2.) Wenn $R_j \in RS$, $typ_{RS_{OS}}(R_j)=(k_1...k_n)$ und $t_1...t_n \in TERM_{SIG_{OS}}$ mit $t_x \in TERM_{k_x}$ und $x=1,...,n$ gelten, dann gilt auch $R_j(t_1,...,t_n) \in FORM_{SIG_{OS}}$. Bei dem Ausdruck $R_j(t_1,...,t_n)$ handelt es sich um eine *atomare ontologische Formel*.

(3.) Wenn F eine ontologische Formel ist, dann ist $\neg F$ eine *zusammengesetzte ontologische Formel*.

---

1) Das Maximalkonzept $\top$ und das Minimalkonzept $\perp$ sind weitere Elemente der Konzeptmenge K, die allerdings für die hier diskutierte Thematik keine Relevanz besitzen.

(4.) Wenn F_1 und F_2 ontologische Formeln sind, dann sind $F_1 \wedge F_2$, $F_1 \vee F_2$, $F_1 \rightarrow F_2$ und $F_1 \leftrightarrow F_2$ *zusammengesetzte ontologische Formeln*.

(5.) Wenn F_1 eine ontologische Formel ist und $x \in VAR$ eine Variable ist, dann sind $\forall x{:}F_1$, $\exists x{:}F_1$ und $\underline{\exists} x{:}F_1$ *zusammengesetzte ontologische Formeln*.

Die Grammatik zur Konstruktion ontologischer Formeln ist strukturell mit der Grammatik zur Konstruktion sortierter Formeln gleich. Ein vermeintlicher Unterschied zwischen den beiden induktiven Grammatiken liegt in einer der Induktionsbasen. Während für atomare sortierte Formeln Relationssymbole verwendet werden, die mit *Sorten* typisiert werden, werden für atomare ontologische Formeln Relationssymbole benötigt, die mit *Konzepten* typisiert werden. Auf die Synonymie der beiden Bezeichner „Sorte“ und „Konzept“ wird in Abschnitt 3.1.3.2.1.1 eingegangen. Entsprechend können in der Grammatik zur Konstruktion von ontologischen Formeln keine formalen Unterschiede gegenüber der Grammatik zur Konstruktion von sortierten Formeln ausgemacht werden.

Die wesentliche Besonderheit ontologischer Formeln im Vergleich zu sortierten Formeln ist die Berücksichtigung der Struktur der Familie $TERMF_{SIG_{OS}}$ aller konzeptspezifischen Termmengen. Für sortierte Formeln ist es – aufgrund der Unstrukturiertheit der Familie $TERMF_{SIG_{SS}}$ aller sortenspezifischer Termmengen – nur möglich, an den entsprechenden Argumentstellen Terme einzusetzen, die originär den sortenspezifischen Termmengen zugeordnet sind, die in der Typisierungsfunktion $typ_{RS_{SS}}$ angegeben sind. Für die Konstruktion ontologischer Formeln können hingegen Terme eingesetzt werden, die zu der jeweiligen konzeptspezifischen Termmenge einer Argumentstelle entweder originär oder – über die Subkonzept- oder Äquivalenzrelation $\sqsubseteq$ bzw. $\triangleq$ vermittelt – derivativ zugeordnet sind. Entsprechend können an einer Argumentstelle keine Terme eingesetzt werden, wenn sie zu konzeptspezifischen Termmengen gehören, die aufgrund der Inkompatibilitätsrelation $\curlyvee$ zu der Termmenge, die für die jeweilige Stelle in der Typisierungsfunktion $typ_{RS_{OS}}$ angegeben ist, disjunkt sein muss.

Diese Besonderheit ontologischer Formeln gegenüber sortierten Formeln braucht allerdings bei der Konstruktion der Erstgenannten nicht berücksichtigt zu werden. Die Besonderheiten werden nämlich bereits bei der Konstruktion von Termen über ontologischen Signaturen berücksichtigt. Durch die Grammatik zur Konstruktion von Termen über ontologischen Signaturen wird eine strukturierte Familie $TERMF_{SIG_{OS}}$ konzeptspezifischer Termmengen erzeugt, in der sowohl die originäre als auch die derivative Zuordnung von Termen zu Konzepten bereits berücksichtigt ist.

Schließlich besteht ein Unterschied zwischen ontologischen Formeln einerseits sowie konventionellen und sortierten Formeln andererseits, der aus der Wertigkeit von Termen in den Formelargumenten herrührt. Sowohl in konventionellen als auch in sortierten Formeln sind als Formelargumente nur solche Terme zugelassen, die zu einwertigen formalen Objekten ausgewertet werden können. Ontologische Formeln können hingegen in ihren Argumenten auch mengenwertige Terme haben. Solche Terme müssen zu formalen Objekten mit Mengencharakter ausgewertet werden. Auf diesen Aspekt ontologischer Terme wird im Rahmen der extensionalen Semantik ontologischer Ausdrücke eingegangen.

Anhand des Potenzials, in ihren Argumenten auch mengenwertige Terme verwenden zu können, wird durch ontologische Formeln eine Ausdrucksmächtigkeit erschlossen, die für prädikatenlogische Formeln nicht üblich ist.[1)] Diese Ausdrucksmächtigkeit erlaubt die Konstruktion formelartiger Aussagen, in denen auch auf *Mengen* von Individuen Bezug genommen wird. Bei einer rein syntaktischen Betrachtung ist dieser Unterschied nicht auf Anhieb auszumachen. Beispielsweise kann bei einer ontologischen Formel der Art $R_j(t_1,t_2,t_3,t_4)$ kein Unterschied zu einer konventionellen oder sortierten Formel ausgemacht werden. Die Unterschiede treten erst bei einer semantischen Analyse ontologischer Formeln hervor. Die o.a. Formel könnte in einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ beispielsweise durch eine Relation der Art

$$r_j(ob_1,\{ob_{2_1},...,ob_{2_n}\},ob_3,ob_4)$$

ausgewertet werden, wenn die Typisierung des Relationssymbols R_j an der zweiten Stelle ein mengenwertiges Konzept aufweist. Auf diesen Aspekt ontologischer Formeln wird später – im Kontext ihrer Auswertung – näher eingegangen.

Formeln, die über ontologischen Signaturen konstruierbar sind, lassen sich ebenso klassifizieren wie Formeln, die über konventionellen oder sortierten Signaturen konstruierbar sind. Hinsichtlich der *Zusammengesetztheit* von Formeln über ontologischen Signaturen wird die Menge $FORM_{SIG_{OS}}$ in die zueinander disjunkten Mengen $AF_{SIG_{OS}}$ und $ZF_{SIG_{OS}}$ unterteilt:

$$FORM_{SIG_{OS}} = AF_{SIG_{OS}} \cup ZF_{SIG_{OS}}$$
$$\text{mit } AF_{SIG_{OS}} \cap ZF_{SIG_{OS}} = \varnothing.$$

Die Bestimmung der Zusammengesetztheit von Formeln über ontologischen Signaturen erfolgt mittels der Funktion

$$tf_{OS}: FORM_{SIG_{OS}} \rightarrow pot_+(FORM_{SIG_{OS}}).$$

Sie weist jeder Formel aus der Menge $FORM_{SIG_{OS}}$ die Menge von Formeln zu, deren Elemente sie als Teilformel in folgender Form zu hat:

(1.) $tf_{OS}(w) = \{w\}$,

(2.) $tf_{OS}(f) = \{f\}$,

(3.) $tf_{OS}(R_j(t_1,...,t_n)) = \{R_j(t_1,...,t_n)\}$,

(4.) $tf_{OS}(\neg F) = \{\neg F\} \cup tf_{OS}(F)$,

(5.) $tf_{OS}((F_1 \bullet F_2)) = \{(F_1 \bullet F_2)\} \cup tf_{OS}(F_1) \cup tf_{OS}(F_2)$ für $\bullet \in \{\wedge,\vee,\underline{\vee},\rightarrow,\leftrightarrow\}$ und

(6.) $tf_{OS}(\bullet x{:}F) = \{\bullet x{:}F\} \cup tf_{OS}(F)$ für $\bullet \in \{\forall,\exists,\underline{\exists}\}$.

---

1) Zu Ausnahmen vgl. ABITEBOUL/GRUMBACH (1991), S. 8 ff.; DOVIER ET AL. (2000), S. 870 ff.; KUPER (1987), S. 13 ff.; KUPER (1988), S. 10 ff.

Die Mengen $AF_{SIG_{OS}}$ und $ZF_{SIG_{OS}}$ lassen sich nun bestimmen als:

$$AF_{SIG_{OS}} = \{F \mid F \in FORM_{SIG_{OS}} \wedge tf_{OS}(F) = \{F\}\}$$
$$\text{und } ZF_{SIG_{OS}} = \{F \mid F \in FORM_{SIG_{OS}} \wedge tf_{OS}(F) \supset \{F\}\}.$$

Die Menge $AF_{SIG_{OS}}$ umfasst demnach alle atomaren Formeln aus der Formelmenge $FORM_{SIG_{OS}}$. Es handelt sich hierbei nur um solche Formeln, die entsprechend den Regeln (1.) bis (3.) der Grammatik zur Konstruktion von Formeln über ontologischen Signaturen konstruiert sind. Entsprechend sind die Formeln, die nach den Regeln (4.) bis (6.) der Grammatik zur Konstruktion von Formeln über ontologischen Signaturen konstruiert sind, stets zusammengesetzte Formeln und werden von der Menge $ZF_{SIG_{OS}}$ umfasst.

Hinsichtlich des Variablenanteils wird die Menge $FORM_{SIG_{OS}}$ in die zueinander disjunkten Mengen $GF_{SIG_{OS}}$ und $VF_{SIG_{OS}}$ unterteilt:

$$FORM_{SIG_{OS}} = GF_{SIG_{OS}} \cup VF_{SIG_{OS}}$$
$$\text{mit } GF_{SIG_{OS}} \cap VF_{SIG_{OS}} = \varnothing.$$

Der Variablenanteil einer ontologischen Formel $F \in FORM_{SIG_{OS}}$ wird mit Hilfe der Funktion

$$var_{F_{OS}}: FORM_{SIG_{OS}} \rightarrow pot(VAR)$$

bestimmt. Sie ordnet jeder ontologischen Formel F eine Menge von Variablen zu. Ihre Bilder sind für alle Formeln $F, F_1, F_2 \in FORM_{SIG_{OS}}$ wie folgt definiert:

(1.) $var_{F_{OS}}(w) = \varnothing$,

(2.) $var_{F_{OS}}(f) = \varnothing$,

(3.) $var_{F_{OS}}(R_j(t_1,...,t_n)) = var_{T_{OS}}(t_1) \cup ... \cup var_{T_{OS}}(t_n)$
für alle $R_j \in RS$ mit $typ_{RS_{OS}}(R_j) = (k_1...k_n)$ und $t_x \in TERM_{k_x}$ für $x=1,...,n$,

(4.) $var_{F_{OS}}(\neg F) = var_{F_{OS}}(F)$
für alle $F \in FORM_{SIG_{OS}}$,

(5.) $var_{F_{OS}}(F_1 \bullet F_2) = var_{F_{OS}}(F) \cup var_{F_{OS}}(F_2)$
für alle $F_1, F_2 \in FORM_{SIG_{OS}}$ und $\bullet \in \{\wedge, \vee, \underline{\vee}, \rightarrow, \leftrightarrow\}$ und

(6.) $var_{F_{OS}}(\bullet{:}F) = var_{F_{OS}}(F) \cup \{x\}$
für alle $F \in FORM_{SIG_{OS}}$ und $\bullet \in \{\forall x, \exists x, \underline{\exists} x\}$.

Die Mengen GF_{OS} und VF_{OS} lassen sich nun bestimmen als:

$$GF_{SIG_{OS}} = \{F \mid F \in FORM_{SIG_{OS}} \wedge var_{F_{OS}}(F) = \varnothing\}$$
$$\text{und } VF_{SIG_{OS}} = \{F \mid F \in FORM_{SIG_{OS}} \wedge var_{F_{OS}}(F) \supset \varnothing\}.$$

Die dritte Klassifikation von Formeln über ontologischen Signaturen erfolgt in Abhängigkeit des Anteils freier Variablen, die in der Formel vorkommen. Demnach lässt sich die Menge $FORM_{SIG_{OS}}$ in die zueinander disjunkten Mengen $OF_{SIG_{OS}}$ und $CF_{SIG_{OS}}$ unterteilen:

$$FORM_{SIG_{OS}} = OF_{SIG_{OS}} \cup CF_{SIG_{OS}}$$
$$\text{mit } OF_{SIG_{OS}} \cap CF_{SIG_{OS}} = \varnothing.$$

Die Bestimmung des Anteils freier Variablen in einer ontologischen Formel erfolgt durch die Funktion

$$fvar_{OS}: FORM_{SIG_{OS}} \rightarrow pot(VAR).$$

Sie ordnet einer ontologischen Formel ihren Anteil freier Variablen wie folgt zu:

(1.) $fvar_{OS}(w) = \varnothing$,

(2.) $fvar_{OS}(f) = \varnothing$,

(3.) $fvar_{OS}(R_j(t_1,...,t_n)) = var_{T_{OS}}(t_1) \cup ... \cup var_{T_{OS}}(t_n)$
für alle $R_j \in RS$ mit $typ_{RS_{OS}}(R_j)=(k_1...k_n)$ und $t_x \in TERM_{k_x}$ für alle $x=1,...,n$,

(4.) $fvar_{OS}(\neg F) = fvar_{OS}(F)$
für alle $f \in FORM_{SIG_{OS}}$,

(5.) $fvar_{OS}(F_1 \bullet F_2) = fvar_{OS}(F_1) \cup fvar_{OS}(F_2)$
für alle $F_1,F_2 \in FORM_{SIG_{OS}}$ und $\bullet \in \{\wedge,\vee,\underline{\vee},\rightarrow,\leftrightarrow\}$ und

(6.) $fvar_{OS}(\bullet F) = fvar_{OS}(F) \backslash \{x\}$
für alle $F \in FORM_{SIG_{OS}}$ und $\bullet \in \{\forall x, \exists x, \underline{\exists} x\}$.

Die Mengen $OF_{SIG_{OS}}$ und $CF_{SIG_{OS}}$ werden mit Hilfe der Funktion $fvar_{OS}$ wie folgt bestimmt:

$$OF_{SIG_{OS}} = \{F \mid F \in FORM_{SIG_{SS}} \wedge fvar_{OS}(F) \supset \varnothing\}$$
$$\text{und } CF_{SIG_{OS}} = \{ F \mid F \in FORM_{SIG_{SS}} \wedge fvar_{OS}(F) = \varnothing\}.$$

Demnach umfasst die Menge $OF_{SIG_{OS}}$ alle Formeln über einer ontologischen Signatur SIG_{OS}, in denen mindestens eine freie Variable vorkommt. Sie werden als *offene* ontologische Formeln bezeichnet. Die Menge $CF_{SIG_{OS}}$ umfasst die dazu komplementäre Menge aller Formeln über einer ontologischen Signatur SIG_{OS}, in denen keine freie Variable vorkommt. Die Elemente von $CF_{SIG_{OS}}$ werden als *geschlossene* ontologische Formeln bezeichnet. Da in Formeln der Menge $GF_{SIG_{OS}}$ grundsätzlich keine Variablen vorkommen, ist die Menge $GF_{SIG_{OS}}$ eine Teilmenge der Menge $CF_{SIG_{OS}}$ aller geschlossenen ontologischen Formeln:

$$GF_{SIG_{OS}} \subseteq CF_{SIG_{OS}}.$$

3.1.3 Semantische Aspekte von Ontologien

3.1.3.1 Aspekte der extensionalen Semantik ontologischer Signaturen

3.1.3.1.1 SIG_{OS}-Strukturen

Die extensionale Semantik ontologischer Signaturen wird – wie bei konventionellen und sortierten Signaturen auch – durch eine *Struktur* gegeben. Eine SIG_{OS}-Struktur $A_{SIG_{OS}}$ zu einer ontologischen Signatur SIG_{OS} ist definiert als:

$$A_{SIG_{OS}}=(OBF_{OS},OPF,RF,IF_{OS}).$$

Die Komponenten einer ontologischen Struktur $A_{SIG_{OS}}$ sind:

(1.) Eine Familie $OBF_{OS}=(OB_k)_{k\in K}$ von konzeptspezifischen Objektmengen, für die gilt:
(1.1.) $OB = \cup_{k\in K} OB_k$,
(1.2.) $\forall k\in K_{EW}$: $(OB_{MEN(k)}=pot(OB_k))$,
(1.3.) $\forall k_1,k_2\in K$: $(k_1 \sqsubseteq k_2) \rightarrow (OB_{k_1} \subseteq OB_{k_2})$,
(1.4.) $\forall k_1,k_2\in K$: $(k_1 \triangleq k_2) \rightarrow (OB_{k_1} = OB_{k_2})$,
(1.5.) $\forall k_1,k_2\in K$: $(k_1 \curlyvee k_2) \rightarrow (OB_{k_1} \cap OB_{k_2} = \varnothing)$,

(2.) eine Familie $OPF=(o_1,...,o_I)$ von *Operationen* o_i mit $i=1,...,I$ und $I\in\mathbb{N}$,

(3.) eine Familie $RF=(r_1,...,r_J)$ von *Relationen* r_j mit $j=1,...,J$ und $J\in\mathbb{N}$ und

(4.) eine Familie $IF_{OS}=(I_K,I_{OPS},I_{RS})$ von Interpretationsfunktionen.

Durch eine SIG_{OS}-Struktur $A_{SIG_{OS}}$ wird eine ontologische Signatur SIG_{OS} extensional interpretiert. Die extensionale Interpretation erfolgt mit den Mitgliedern der Familie $IF_{OS}=(I_K,I_{OPS},I_{RS})$ aller (extensionalen) Interpretationsfunktionen. Die Interpretationsfunktionen aus IF_{OS} werden dazu verwendet, den objektsprachlichen Komponenten der ontologischen Signatur SIG_{OS} Konstrukte aus der SIG_{OS}-Struktur $A_{SIG_{OS}}$ zuzuordnen.

Mit der Interpretationsfunktion

$$I_K: K \rightarrow OBF_{OS}$$

wird jedes Konzept $k\in K$ aus einer ontologischen Signatur SIG_{OS} durch eine konzeptspezifische Objektmenge $OB_k\in OBF_{OS}$ extensional interpretiert. Die konzeptspezifische Objektmenge OB_k, mit der ein Konzept $k\in K$ extensional interpretiert wird, wird auch als dessen *Extension* bezeichnet. Die formalen Objekte $ob_1,...,ob_n\in OB_k$ werden als *Individuen* bezeichnet.[1)] Jedes Individuum ob_u aus einer k-spezifischen Objektmenge OB_k

1) Je nach Argumentationskontext wird im Folgenden in Bezug auf ein formales Objekt ob_u wahlweise entweder die Bezeichnung „Individuum“ oder die dazu übergeordnete Bezeichnung „formales Objekt“ verwendet.

ist eine *Instanz* des Konzepts k. Die Zuordnung eines Individuums ob_u zur konzeptspezifischen Objektmenge OB_k wird entsprechend auch als *Instanziierung* des Konzepts k durch das Individuum ob_u bezeichnet.

Es wird zudem

$$\forall k \in K: OB_k \neq \varnothing$$

eingefordert. Insofern dürfen in einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ keine leeren konzeptspezifischen Objektmengen vorkommen. Diese Festlegung ist notwendig, um den Bildbereich von Termauswertungsfunktionen bestimmen zu können.[1)] Es kann beispielsweise festgelegt werden, dass in jeder konzeptspezifischen Objektmenge OB_k mindestens ein ausgezeichnetes formales Objekt ob_{nil} enthalten sein muss. Dieses ausgezeichnete formale Objekt übernimmt die Funktion eines „Pointers" auf jede „eigentlich" leere konzeptspezifische Objektmenge.

Die Vereinigung aller konzeptspezifischen Objektmengen entspricht dem prädikatenlogischen Universum OB. Das Universum OB umfasst alle formalen Objekte aus allen konzeptspezifischen Objektmengen. Somit entspricht OB der Vereinigung aller konzeptsspezifischen Objektmengen:

$$OB = \cup_{k \in K} OB_k.$$

Für SIG_{SS}-Strukturen zu einer sortierten Signatur SIG_{SS} sind keine weiteren Differenzierungen außerhalb der sortenspezifischen Zuordnung von Individuen vorgesehen. In SIG_{OS}-Strukturen zu ontologischen Signaturen SIG_{OS} kann hingegen mit Hilfe der metasprachlichen Strukturierungsrelationen $\sqsubseteq$, $\triangleq$ und $\curlyvee$ eine dreifache Differenzierung und Strukturierung der Familie OBF_{OS} vorgenommen werden. Hinsichtlich der Subkonzeptrelation $\sqsubseteq$ wird eine *Verschachtelung* der konzeptspezifischen Objektmengen aus OBF_{OS} konstruiert. Wenn für ein Konzept k_1 gilt, dass es in Beziehung ($k_1 \sqsubseteq k_2$) zu einem zweiten Konzept k_2 steht, muss die konzeptspezifische Objektmenge OB_{k_1} eine Teilmenge der konzeptspezifischen Objektmenge OB_{k_2} enthalten sein. Auf die Eigenarten der Subkonzeptrelation $\sqsubseteq$ wird in Abschnitt 3.1.3.2.2.1 näher eingegangen.

Während mit der Subkonzeptrelation $\sqsubseteq$ eine *vertikale* Anordnung von Konzepten bewirkt wird, werden die metasprachlichen Relationen $\triangleq$ und $\curlyvee$ dazu verwendet, Konzepte in *horizontaler* Weise miteinander in Beziehung zu setzen. Mit Hilfe der Äquivalenzrelation $\triangleq$ wird ausgedrückt, dass zwei konzeptspezifische Objektmengen OB_{k_1} und OB_{k_2} gleich sein müssen, wenn für die Konzepte $k_1 \triangleq k_2$ gilt. Den entgegengesetzten Fall stellt die Inkompatibilitätsrelation $\curlyvee$ dar. Wenn für zwei Konzepte k_1 und k_2 die Beziehung ($k_1 \curlyvee k_2$) gilt, müssen die jeweiligen konzeptspezifischen Objektmengen OB_{k_1} und OB_{k_2} zueinander disjunkt sein. Auf die Äquivalenzrelation $\triangleq$ und die Inkompatibilitätsrelation $\curlyvee$ wird in den Abschnitten 3.1.3.2.2.1 näher eingegangen.

Mit den zuvor aufgeführten metasprachlichen Relationen wird den Mitgliedern der Mengenfamilie OBF_{OS} eine Differenzierung und Strukturierung aufgelagert. Darüber

1) vgl. Abschnitt 3.1.3.1.2.1 .

hinaus können auch die Individuen aus den konzeptspezifischen Objektmengen hinsichtlich zweier Dimensionen differenziert werden. Es handelt sich dabei um die Dimensionen der *Mengenwertigkeit* und der *Zusammengesetztheit*.

Hinsichtlich ihrer Mengenwertigkeit werden alle Individuen aus OB_{OS} in *einwertige* und *mengenwertige* Individuen unterschieden. Als einwertige Individuen werden solche formalen Objekte aus OB_{OS} bezeichnet, die Skalare darstellen. Mengenwertige Individuen sind dagegen formale Objekte, die Mengen darstellen, deren Elemente wiederum stets einwertige[1)] Individuen sein müssen. Auf diese Differenzierung formaler Objekte wird in Abschnitt 3.1.3.2.1.1 näher eingegangen.

Hinsichtlich ihrer Zusammengesetztheit können die Elemente der Menge OB_{OS} in *atomare* und *zusammengesetzte* Individuen unterschieden werden.[2)] Bei atomaren Individuen handelt es sich um solche formalen Objekte, die als Elemente einer konzeptspezifischen Objektmenge originär eingeführt werden. *Zusammengesetzte* oder *komplexe* Individuen sind hingegen solche formalen Objekte, die aus der Anwendung einer Operation $o_i \in OPF_{OS}$ mit der Funktionsvorschrift o_i: $OB_{k_1} \times .... \times OB_{k_n} \rightarrow OB_k$ auf ein n-Tupel $(ob_1,...,ob_n)$ von Individuen mit $ob_x \in OB_{k_x}$ für x=1,...,n hervorgehen. Bei den Individuen im Argument des Ausdrucks $o_i(ob_1,...,ob_n)$ kann es sich wiederum um atomare oder zusammengesetzte formale Objekte handeln.

Die Mengenfamilien OPF und RF enthalten jeweils Operationen bzw. Relationen, mit denen die jeweiligen Konstrukte aus einer ontologischen Signatur SIG_{OS} extensional interpretiert werden. Dabei interpretieren die Interpretationsfunktionen

$$I_{OPS}: OPS \rightarrow OPF$$
$$\text{und } I_{RS}: RS \rightarrow RF$$

jeweils auf extensionale Weise ontologische Operations- bzw. Relationssymbole. Mit der Interpretationsfunktion I_{OPS} wird jedes Operationssymbol

$$O_i \in OPS \text{ mit } typ_{OPS_{OS}}(O_i)=(k_1...k_n,k)$$

durch eine Operation

$$I_{OPS}(O_i)=o_i \text{ mit } o_i: OB_{k_1} \times ... \times OB_{k_n} \rightarrow OB_k$$

---

1) Mengenwertige Individuen, die sich wiederum aus mengenwertigen Individuen zusammensetzen, sind zwar nicht Betrachtungsgegenstand der vorliegenden Arbeit, können aber ohne komplexe Ausweitungen des Kalküls aufgenommen werden. Die einzige Variation des Kalküls, die noch zu ergänzen wäre, läge in einer Änderung des Vorbereichs der Funktion MEN, mit der einwertigen Konzepten mengenwertige Konzepte zugeordnet werden. Wenn durch eine Variation des Vorbereichs der Funktion MEN auch die Generierung mengenwertiger Konzepte von mengenwertigen Konzepten zugelassen wird, können Letztgenannte durch mengenwertige Individuen extensional interpretiert werden, die sich selbst auch aus mengenwertigen Individuen zusammensetzen.

2) Vgl. ZELEWSKI (1995), Bd. 4, S. 17.

extensional interpretiert. Analog zu den Extensionen von Konzepten wird die Operation $o_i \in OPF$, mit der das Operationssymbol O_i extensional interpretiert wird, als dessen *Extension* bezeichnet.[1)]

Der Vorbereich der Operation $I_{OPS}(O_i)=o_i$, mit der das Operationssymbol O_i extensional interpretiert wird, besteht aus dem kartesischen Produkt konzeptspezifischer Objektmengen. Die konzeptspezifischen Objektmengen sind genau den Konzepten zugeordnet, mit denen das Operationssymbol O_i an den jeweiligen Argumentstellen typisiert ist.

Ontologische Operationssymbole, die in der Form $typ_{OPS_{OS}}(O_i)=(\lambda,k)$ typisiert sind,[2)] werden durch Operationen mit der Operationsvorschrift $o_i{:}\rightarrow OB_k$ extensional interpretiert. Solche Operationen, die im Vorbereich ein „null-faches" kartesisches Produkt aufweisen, werden als *Konstanten* bezeichnet. Konstanten sind solche Operationen, deren „Anwendung" auf ein leeres Argument ein Individuum $o_i()=ob_u$ hervorbringt.

Jedes Relationssymbol

$$R_j \in RS \text{ mit } typ_{RS_{OS}}(R_j)=(k_1 ... k_n)$$

wird durch eine Relation

$$I_{RS}(R_j)=r_j \text{ mit } r_j \subseteq OB_{k_1} \times ... \times OB_{k_n}$$

extensional interpretiert. Die Relation $r_j \in RF$, mit der das Relationssymbol $R_j \in RS$ extensional interpretiert wird, wird auch als dessen *Extension* bezeichnet.[3)] Jede Extension $r_j \in RF$ zu einem Relationssymbol $R_j \in RS$ umfasst jeweils n-Tupel $ob_1,...,ob_n$ von Individuen, wobei für jedes Individuum ob_u und für $typ_{RS_{OS}}(R_j)=(k_1,...,k_n)$ mit $1 \leq u \leq n$ gilt: $ob_u \in OB_{k_u}$.

Der Argumentbereich der Relation $r_j=I_{RS}(R_j)$, mit der das Relationssymbol R_j extensional interpretiert wird, besteht aus dem kartesischen Produkt konzeptspezifischer Objektmengen. Die jeweiligen konzeptspezifischen Objektmengen sind genau den Konzepten zugeordnet, mit denen das Relationssymbol R_j an den jeweiligen Stellen typisiert ist. Relationssymbole, die in der Form $typ_{RS_{OS}}(R_j)=\lambda$ typisiert sind, wurden in ontologischen Signaturen ausgeschlossen, wodurch der aussagenlogische Grenzfall für Ontologien nicht in Betracht gezogen wird.

Eine SIG_{OS}-Struktur $A_{SIG_{OS}}$ liefert *eine* mögliche extensionale Interpretation einer ontologischen Signatur SIG_{OS}. Die ontologische Signatur SIG_{OS} kann jedoch auch durch weitere SIG_{OS}-Strukturen

1) Verwechslungen sind nicht zu befürchten, da der Begriff *Extension* immer in Zusammenhang mit dem jeweiligen Konstrukt verwendet werden wird, dessen extensionale Interpretation es darstellt.

2) Operationssymbole, die in ihrem Vorbereich mit dem Symbol λ typisiert sind, wurden bereits früher als ontologische *Konstantensymbole* eingeführt.

3) Auch hierbei sind Verwechslungen nicht zu befürchten, da der Begriff *Extension* immer in Zusammenhang mit dem jeweiligen Konstrukt verwendet werden wird, dessen extensionale Interpretation es darstellt.

$$A_{SIG_{OS.1}}=(OBF_{OS_1},OPF_1,RF_1,I_{OS_1}) , \dots , A_{SIG_{OS.X}}=(OBF_{OS_X},OPF_X,RF_X,I_{OS_X})$$

extensional interpretiert werden. Jede SIG_{OS}-Struktur $A_{SIG_{OS.x}}$ mit x=1,...,X verfügt über eine Familie OBF_{OS_x} von konzeptspezifischen Objektmengen, durch deren Mitglieder die Konzepte aus SIG_{OS} extensional interpretiert werden. Darüber hinaus verfügt jede SIG_{OS}-Struktur $A_{SIG_{OS.x}}$ über eine Familie OPF_x von Operationen und eine Familie RF_x von Relationen. Schließlich ist in jeder SIG_{OS}-Struktur $A_{SIG_{OS.x}}$ eine Familie I_{OS_x} bijektiver Interpretationsfunktionen enthalten. Die Menge $A(SIG_{OS})$ umfasst alle SIG_{OS}-Strukturen, die für eine solche extensionale Interpretation der ontologischen Signatur SIG_{OS} in Frage kommen.

3.1.3.1.2 Auswertung von ontologischen Ausdrücken

3.1.3.1.2.1 Auswertung von ontologischen Termen

Mit den Interpretationsfunktionen I_K, I_{OPS} und I_{RS} sind Mittel vorgestellt worden, mit denen deskriptiven Symbolen und Konzepten ontologischer Signaturen Komponenten von SIG_{OS}-Strukturen zugeordnet werden können. Komplementär zur extensionalen Interpretation der deskriptiven Symbole und Konzepte einer ontologischen Signatur ist die extensionale Interpretation von *Ausdrücken* über ontologischen Signaturen notwendig, die mit den logischen und deskriptiven Symbolen aus dem formalsprachlichen Alphabet $ALPH_{OS}$, das jeder ontologischen Signatur SIG_{OS} zugrunde liegt, konstruiert wurden. Entsprechend der Unterteilung der Menge $EXPR_{SIG_{OS}}$ aller Ausdrücke über einer ontologischen Signatur SIG_{OS} in die Menge $TERM_{SIG_{OS}}$ aller Terme und die Menge $FORM_{SIG_{OS}}$ aller Formeln wird zunächst die extensionale Interpretation von Termen über ontologischen Signaturen vorgestellt. Im Anschluss wird auf die extensionale Interpretation von Formeln über ontologischen Signaturen eingegangen.

Die extensionale Interpretation von Termen über ontologischen Signaturen wird – analog zu der extensionalen Interpretation von Ausdrücken über konventionellen oder sortierten Signaturen – als *Auswertung* bezeichnet. Die Auswertung von ontologischen Termen erfolgt durch die Mitglieder einer Familie

$$ITF_{OS}=(IT_k)_{k\in K}$$

konzeptspezifischer Termauswertungsfunktionen

$$IT_k\colon TERM_k \rightarrow OB_k \text{ für } k\in K.$$

Jede konzeptspezifische Termauswertungsfunktion IT_k ordnet allen Termen $t_1...t_n\in TERM_k$ aus einer k-spezifischen Termmenge $TERM_k$ jeweils ein Individuum $ob_1,...,ob_n$ aus der k-spezifischen Objektmenge OB_k zu. Somit werden alle Terme über einer ontologischen Signatur SIG_{OS} extensional durch Individuen aus einer SIG_{OS}-Struktur $A_{SIG_{OS}}\in A(SIG_{OS})$ interpretiert.

Die Auswertung von Termen über ontologischen Signaturen hat mit der Auswertung von Termen über sortierten Signaturen gemeinsam, dass sie *strukturiert* erfolgt. Dies unterscheidet beide Auswertungsweisen von der Art der Auswertung von Termen über konventionellen Signaturen. Die Termauswertungsfunktion IT_k kann nämlich nur auf Terme aus einer solchen Termmenge IT_k angewendet werden, die dem jeweiligen Konzept k zugeordnet ist. Entsprechend können in den Bildbereichen konzeptspezifischer Termauswertungsfunktionen nur Individuen aus solchen Objektmengen vorkommen, die auch dem Konzept k zugeordnet sind. Beides wird durch die Identität der Indizes bei der Definition konzeptspezifischer Termauswertungsfunktionen gewährleistet.

Trotz der vielfachen Gemeinsamkeiten weist die Art der Auswertung von Termen über ontologischen Signaturen auch Unterschiede zu der Art der Auswertung von Termen über sortierten Signaturen auf. Die Unterschiede sind auf die metasprachlichen Strukturierungsrelationen $\sqsubseteq$, $\triangleq$ und $\curlyvee$ zurückzuführen. Wie bereits bei der Definition von Termen über ontologischen Signaturen dargestellt, werden die Mitglieder der Familie $TERMF_{SIG_{OS}}$ aller konzeptspezifischen Termmengen über einer ontologischen Signatur SIG_{OS} mit den Strukturierungsrelationen zueinander in mengentheoretische Beziehungen gesetzt. Die innere Struktur, die die Mengenfamilie $TERMF_{SIG_{OS}}$ durch die Strukturierungsrelationen $\sqsubseteq$, $\triangleq$ und $\curlyvee$ aufgelagert bekommt, hat Auswirkungen auf die Bild- und Definitionsmengen konzeptspezifischer Termauswertungsfunktionen.

Durch eine konzeptspezifische Termauswertungsfunktion IT_k können nicht nur solche Terme ausgewertet werden, die dem Konzept k originär zugeordnet sind, sondern auch alle Terme, die z.B. aufgrund einer Subkonzeptbeziehung ($k_n \sqsubseteq k$) einer Termmenge $TERM_{k_n}$ mit $TERM_{k_n} \subseteq TERM_k$ entstammen und somit dem Konzept k derivativ zugeordnet sind. Es erfolgt keine Differenzierung, ob ein Term t dem Konzept k originär oder derivativ zugeordnet ist. Eine konzeptspezifische Termauswertungsfunktion IT_k kann in ihrem Definitionsbereich auch solche ontologischen Terme aufweisen, die dem Konzept k derivativ zugeordnet sind.

Analog verhält es sich mit der Äquivalenzrelation $\triangleq$. Die konzeptspezifische Termauswertungsfunktion IT_k kann in ihrem Definitionsbereich auch solche Terme aus allen Termmengen $TERM_{k_n}$ aufweisen, die nicht originär dem Konzept k zugeordnet sind, sondern aufgrund der Äquivalenzbeziehung $k_n \triangleq k$ zu der Termmenge $TERM_k$ des Konzepts k gezählt werden. Es gelten nämlich $TERM_{k_n}$=$TERM_k$ für alle Konzepte $k_n,k \in K$ mit $k_n \triangleq k$. Diese Anforderung an die Struktur konzeptspezifischer Termmengen wurde bereits bei der Definition von Termen über ontologischen Signaturen eingeführt.

Ein Ausschluss von Termen im Definitionsbereich konzeptspezifischer Termauswertungsfunktionen wird durch die Inkompatibilitätsrelation $\curlyvee$ bewirkt. Wenn nämlich $k_1 \curlyvee k_2$ gilt, darf kein Term aus $TERM_{k_1}$ im Definitionsbereich der konzeptspezifischen Termauswertungsfunktionen IT_{k_2} vorkommen und umgekehrt.[1] Dies ergibt sich aus der Forderung der Disjunktheit zweier konzeptspezifischer Termmengen $TERM_{k_1}$ und $TERM_{k_2}$ bei Inkompatibilität der Konzepte k_1 und k_2.

1) Der wechselseitige Ausschluss der Definitionsbereiche ist durch die Symmetrie der Inkompatibilitätsrelation $\curlyvee$ bedingt.

Wie bei Variablen aus sortierten Signaturen, so ist auch für die Auswertung von ontologischen Termen, die aus Variablen hervorgehen, zunächst die Belegung der Variablen mit Individuen notwendig.[1)] Die Belegung von Variablen aus einer konzeptspezifischen Variablenmenge VAR_k erfolgt durch das jeweilige Mitglied bel_k der Familie $belf_{OS}$ konzeptspezifischer Variablenbelegungsfunktionen:[2)]

$$belf_{OS}=(bel_k)_{k\in K}.$$
$$\text{mit } bel_k\text{: } VAR_k \rightarrow OB_k \text{ für } k\in K.$$

Mit einer konzeptspezifischen Variablenbelegung bel_k wird Variablen aus einer konzeptspezifischen Variablenmenge VAR_k ein Individuum zugeordnet. Das zugeordnete Individuum hat stets einer Objektmenge OB_k zu entstammen, die dem gleichen Konzept k zugeordnet ist, dem auch die Variablenmenge VAR_k zugeordnet ist. Dies wird – analog zu den konzeptspezifischen Termauswertungsfunktionen IT_k – durch die Identität der Indizes bei der Definition konzeptspezifischer Variablenbelegungen gewährleistet. Auch hierbei gilt, dass im Definitionsbereich einer konzeptspezifischen Variablenbelegung bel_k auch solche Terme vorkommen können, die zwar nicht originär, allerdings derivativ dem Konzept k zugeordnet sind.

Für die spätere Auswertung von quantifizierten Formeln ist auch bei Variablen über ontologischen Signaturen eine bedingte Variablenbelegung notwendig.

Jede *bedingte konzeptspezifische Variablenbelegungsfunktion*

$$bel_k[x^*/ob_u](x)=\begin{cases} bel_k(x) & \text{wenn } x \neq x^* \\ ob_u & \text{wenn } x = x^* \end{cases}$$

weist jeder Variablen x aus der konzeptspezifischen Variablenmenge VAR_k ein Individuum aus der konzeptspezifischen Objektmenge OB_k entsprechend der Variablenbelegung bel_k zu, wenn sich die Variable x von einer näher spezifizierten Variable x* unterscheidet. Ansonsten wird x mit dem ausgezeichneten Individuum ob_u belegt.

Die Auswertung von Termen über ontologischen Signaturen ist bei einer Variablenbelegung bel_k wie folgt rekursiv definiert:

(1.) $IT_k(x)=bel_k(x)$
für jede Variable $x\in VAR_k$ und jedes Konzept $k\in K$,

1) Die Notwendigkeit gilt nur so lange, wie die *Substitution* ontologischer Ausdrücke noch nicht eingeführt wurde. Die Substitution ontologischer Ausdrücke wird in Abschnitt 3.1.4.3.1 vorgestellt. Im Rahmen dieses Abschnitts wird auch auf die *Grundsubstitution* eingegangen. Es handelt sich hierbei um ein Verfahren, bei dem alle Variablen aus ontologischen Ausdrücken durch Grundterme ersetzt werden. Der daraus resultierende Ausdruck kann ohne Rückgriff auf eine Variablenbelegung in einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ ausgewertet werden.

2) Anstelle der Bezeichnung *Variablenbelegungsfunktion* wird fortan auch die Bezeichnung *Variablenbelegung* zugelassen.

(2.) $IT_k(O_i)=I_{OPS}(O_i)=o_i$ mit $o_i()=ob_u$ und $ob_u \in OB_k$
für jedes Konstantensymbol $O_i \in OPS$ mit $typ_{OPS_{OS}}(O_i)=(\lambda,k)$ und

(3.) $IT_k(O_i(t_1...t_n))=o_i(IT_{k_1}(t_1),...,IT_{k_n}(t_n))$
für jedes Operationssymbol $O_i \in OPS$ mit $typ_{OPS_{OS}}(O_i)=(k_1...k_n,k_{n+1})$ mit den Konzepten $k_1,...,k_n,k_{n+1} \in K$ und den Termen $t_1,...,t_n$ mit $t_x \in TERM_{k_x}$ für x=1,...,n und $n \in \mathbb{N}_+$.

Aufgrund der Struktur, die der Familie $TERMF_{SIG_{OS}}$ konzeptspezifischer Termmengen durch die Strukturierungsrelationen $\sqsubseteq$ und $\triangleq$ überlagert wird, können dieselben Terme durchaus in unterschiedlichen konzeptspezifischen Termmengen enthalten sein. Entsprechend kann ein Term in den Definitionsbereichen unterschiedlicher konzeptspezifischer Termauswertungsfunktionen vorkommen. Auf diesen Aspekt der Auswertung von ontologischen Termen wurde bereits eingegangen. Zu beachten ist allerdings hierbei, dass die Auswertung eines Terms $t \in TERM_{k_1} \cap TERM_{k_2}$ entsprechend den zwei unterschiedlichen Termauswertungsfunktionen IT_{k_1} und IT_{k_2} nicht zu zwei unterschiedlichen Individuen führen darf. Diese Bedingung an alle Termauswertungsfunktionen wird mit der Integritätsregel

$$\forall k_1,k_2 \in K,\ t \in TERM_{SIG_{OS}}: (IT_{k_1}(t)=ob_1 \land IT_{k_2}(t)=ob_2) \rightarrow ob_1=ob_2$$

erfüllt. Wenn z.B. $k_1 \sqsubseteq k_2$ und demnach $TERM_{k_1} \subseteq TERM_{k_2}$ gelten, müssen die beiden konzeptspezifischen Termauswertungsfunktionen IT_{k_1} und IT_{k_2} einem Term $t \in TERM_{k_1} \cap TERM_{k_2}$ dasselbe Individuum $ob_1=ob_2$ zuweisen. Ansonsten würde der Term t in seiner „Rolle“ als Term zum Konzept k_1 anders interpretiert werden als in seiner „Rolle“ als Term zum Konzept k_2. Dennoch sind die konzeptspezifischen Termauswertungsfunktionen nicht notwendig injektiv. Denn es können sehr wohl zwei unterschiedliche Terme $t_1 \in TERM_{k_1}$ und $t_2 \in TERM_{k_2}$ zu demselben Individuum $IT_{k_1}(t_1)=IT_{k_2}(t_2)=ob_u$ mit $ob_u \in (OB_{k_1} \cap OB_{k_2})$ ausgewertet werden. Dies kann beispielsweise dann der Fall sein, wenn ob_u sowohl der Auswertung eines atomaren Terms (t_1) als auch der Auswertung eines zusammengesetzten Terms (t_2) entspricht.

Das Rekursionsschema für konzeptspezifische Termauswertungsfunktionen deckt sich mit dem Rekursionsschema für sortenspezifische Termauswertungsfunktionen. Demnach wird die Rekursionsbasis durch die Regeln (1.) und (2.) gelegt. Die Auswertung einer Variablen x in ihrer „Rolle“ als Term ist stets genau das formale Objekt ob_u, durch das x entsprechend einer konzeptspezifischen Variablenbelegung bel_k belegt wird. Dadurch, dass einerseits die konzeptspezifische Termauswertungsfunktion IT_k und die konzeptspezifische Variablenbelegung bel_k und andererseits die konzeptspezifische Objektmenge OB_k im Nachbereich von bel_k alle mit demselben Konzept k indiziert sind, wird die Konzepttreue[1)] bei der Auswertung von Termen, die aus Variablen hervorgehen, gewährleistet.

1) Der Begriff *Konzepttreue* ist von dem Begriff *Sortentreue* abgeleitet, der im Rahmen der Auswertung von Termen über sortierten Signaturen eingeführt wurde.

Die zweite Rekursionsbasis ist die Regel (2.), mit der die Auswertung ontologischer Konstantensymbole definiert wird. Die Auswertung eines Konstantensymbols $O_i \in OPS$ mit $typ_{OPS_{OS}}(O_i)=(\lambda,k)$ entspricht einer Konstanten $o_i \in OPF$ mit $o_i: \rightarrow OB_k$ und $o_i()=ob_u$. Somit stimmt die Auswertung eines Konstantensymbols O_i mit dem Individuum ob_u überein, das aus der Anwendung der Extension o_i von O_i auf das leere Argument hervorgeht.

Die beiden Rekursionsbasen beziehen sich auf die Auswertung von atomaren Termen aus der Menge $AT_{SIG_{OS}}$. Es handelt sich hierbei um Terme, die entweder Variablen sind oder Konstantensymbolen entsprechen. Für die Auswertung von zusammengesetzten Termen aus einer konzeptspezifischen Menge ZT_k wird die Regel (3.) benötigt. Sie bildet jeden zusammengesetzten Term $t \in TERM_k$ mit $t=O_i(t_1,...,t_n)$ auf ein Individuum $ob_u \in OB_k$ mit $ob_u=o_i(ob_1,...,ob_n)$ ab. Das Individuum ob_u geht in diesem Fall aus der Anwendung der Operation o_i auf das n-Tupel $(ob_1,...,ob_n)$ von Individuen hervor. Demnach erfolgt die Auswertung von Termen über ontologischen Signaturen *strukturerhaltend*. Atomare Terme werden zu atomaren Individuen und zusammengesetzte Terme zu zusammengesetzten Individuen ausgewertet. Insbesondere erlaubt es das Rekursionsschema aller konzeptspezifischen Termauswertungsfunktionen, dass die Terme in den Argumenten zusammengesetzter Terme selbst auch zusammengesetzt sein dürfen. Die Termauswertungsfunktionen werden in diesem Fall solange auf zusammengesetzte Terme angewendet, bis eine der beiden Rekursionsbasen erreicht ist.

Die Möglichkeit zur Auswertung zusammengesetzter Terme erweist sich als besonders fruchtbar für die Formulierung von *Anfragen* an eine SIG_{OS}-Struktur $A_{SIG_{OS}}$ zu einer ontologischen Signatur SIG_{OS}. Zusammengesetzte Terme lassen sich nämlich dazu verwenden, *Pfadausdrücke* zu formulieren, die auch verschachtelte Anfragen ermöglichen. Pfadausdrücke sind solche Ausdrücke, die über eine Anwendungskette von Operationssymbolen auf Terme definiert sind.[1)] Dieser Zusammenhang wird im Folgenden anhand eines Beispiels verdeutlicht. Dabei werden die folgende ontologische Signatur SIG_{OS} und die SIG_{OS}-Struktur $A_{SIG_{OS}}$ auszugsweise zu Grunde gelegt:

1) Vgl. KIFER ET AL. (1995), S. 817 ff.; UPHOFF (1997), S. 47 für die Umsetzung von Pfadausdrücken in der Wissensrepräsentationssprache F-LOGIC. Vgl. KARVOUNARAKIS ET AL. (2002), S. 596 für die Umsetzung von Pfadausdrücken in der Anfragesprache RQL, die für die Wissensrepräsentationssprache RDF definiert ist.

$\underline{SIG_{OS}}$			
K	=	$\{\top,\bot$,Mann,Frau,Person$\}$;	
$\sqsubseteq$	=	{(Mann,Person),(Frau,Person);	
OPS	=	Arbeitet_fuer:	Person → Unternehmen
		Verheiratet_mit:	Person → Person
		Tochterunt_von:	Unternehmen → Unternehmen;
$\underline{VARF_{OS}}$			
VAR_{Person} =		{x};	
$\underline{A_{SIG_{OS}}}$			
OBF_{OS}		(OB_{Person}	={hans,peter,werner,sabine,elke},
		OB_{Mann}	={hans,peter,werner},
		OB_{Frau}	={sabine,elke},
		$OB_{Unternehmen}$	={arnulf_GmbH,bruederich_AG,bleidorf_KG};
OPF		(verheiratet_mit(sabine)	= hans,
		verheiratet_mit(hans)	= sabine,
		verheiratet_mit(werner)	= elke,
		verheiratet_mit(elke)	= werner,
		arbeitet_fuer(hans)	= arnulf_GmbH,
		arbeitet_fuer(werner)	= arnulf_GmbH,
		tochterunt_von(arnulf_GmbH)	= bruederich_AG).

Der Ausdruck

Tochterunt_von(Arbeitet_fuer(Verheiratet_mit(x)))

liefert bei Belegung der Variablen $x \in VAR_{Person}$ durch $bel_{Person}(x)$=sabine mit sabine$\in OB_{person}$ das Objekt bruederich_AG$\in OB_{Unternehmen}$. Das Vorgehen zur Ermittlung des Pfades ist durch die folgende Tabelle gegeben:

Rekursions-stufe	Terme	Individuen
Stufe 0	$x \in TERM_{Person}$ mit $x \in VAR_{Person}$ und $VAR_{person} \subseteq TERM_{Person}$	$IT_{Person}(x)=bel_{Person}(x)$ $=sabine \in OB_{Person}$
Stufe 1	(Verheiratet_mit(x))$\in TERM_{Person}$	verheiratet_mit(sabine) =hans$\in OB_{Person}$
Stufe 2	(Arbeitet_fuer(y))$\in Term_{Unternehmen}$ mit (y=Verheiratet_mit(x))$\in TERM_{Person}$	arbeitet_fuer(hans) =arnulf_GmbH$\in OB_{Unternehmen}$
Stufe 3	(Tochterunt__von(z))$\in TERM_{Unternehmen}$ mit (z=Arbeitet_fuer(y))$\in TERM_{Unternehmen}$ und (y=Verheiratet_mit(Sabine))$\in TERM_{Person}$	tochterunt_von(arnulf_GmbH) =bruederich_AG$\in OB_{Unternehmen}$

Tabelle 4: Rekursive Konstruktion und Auswertung von Termen

In der Tabelle 4 ist exemplarisch aufgezeigt, wie das „endgültige“ Individuum *bruederich_AG* aus der konzeptspezifischen Objektmenge $OB_{Unternehmen}$ ermittelt werden kann. Die Unterscheidung zwischen der Spalte „Terme“ und der Spalte „Individuen“ spiegelt die konsequente Unterscheidung zwischen den rein syntaktisch definierten Termen über der ontologischen Signatur SIG_{OS} und den semantisch motivierten Individuen aus der SIG_{OS}-Struktur $A_{SIG_{OS}}$ wider. Während die erstgenannte Menge lediglich „abstrakte“ Ausdrücke beinhaltet, umfasst die zweitgenannte Menge „konkrete“ formale Objekte aus Objektmengen und ebenso „konkrete“ Operationen auf den formalen Objekten.

Eine letzte bemerkenswerte Eigenschaft weist die Auswertung ontologischer Terme bezüglich mengenwertiger Terme auf. Mengenwertige Terme werden auch stets zu *mengenwertigen Individuen* ausgewertet. Das muss genau dann der Fall sein, wenn das Konzept k, dem eine konzeptspezifische Termauswertungsfunktion IT_k zugeordnet ist, selbst mengenwertig ist.[1)] Mengenwertige Individuen sind wiederum stets selbst Mengen, so dass die Auswertung eines mengenwertigen Terms zu einer Menge führen muss. Beispielsweise muss ein Term t aus einer konzeptspezifischen Menge $TERM_k$ mengenwertiger Terme für das Konzept k mit $k \in K_{MW}$ zu einem mengenwertigen Individuum $IT_k(t)=ob_u=\{ob_{u_1},...,ob_{u_n}\}$ führen. Ein Term t aus einer konzeptspezifischen Menge $TERM_k$ einwertiger Terme mit $k \in K_{EW}$ muss hingegen zu einem einwertigen Individuum $IT_k(t)=ob_u$ ausgewertet werden.

1) Auf diese Eigenschaft mengenwertiger Individuen wurde zuvor hingewiesen. Er wird in Abschnitt 3.1.3.2.1.1 weiter vertieft. Für Aspekte der Auswertung mengenwertiger Terme reichen allerdings die bisherigen Ausführungen aus.

3.1.3.1.2.2 Auswertung von ontologischen Formeln

Für die vollständige Zusammenführung ontologischer Signaturen und SIG_{OS}-Strukturen wird schließlich die Auswertung von ontologischen Formeln benötigt. Formeln sind neben Termen die zweite Ausdrucksart, die mit den Ausdrucksmitteln einer ontologischen Signatur SIG_{OS} konstruiert werden kann. Wenn die Terme in den Argumenten atomarer Formeln mit Hilfe konzeptspezifischer Termauswertungsfunktionen aus der Familie ITF_{OS} zu Individuen ausgewertet sind, können sowohl atomare als auch zusammengesetzte Formeln über ontologischen Signaturen zu Wahrheitswerten ausgewertet werden.

Das Prinzip zur Auswertung ontologischer Formeln deckt sich mit dem Prinzip zur Auswertung sortierter Formeln. Um die *Bestätigung* einer ontologischen Formel $F \in FORM_{SIG_{OS}}$ unter einer Familie $belf_{OS}$ konzeptspezifischer Variablenbelegungen ausdrücken zu können, wird die Modellrelation $\models$ verwendet:

$$(A_{SIG_{OS}}, belf_{OS}) \models F$$

Dabei ist die Modellrelation $\models$ für Formeln über ontologischen Signaturen wie folgt definiert:

(1.) $(A_{SIG_{OS}}, belf_{OS}) \models w$ gilt für *jedes* Tupel $(A_{SIG_{OS}}, belf_{OS})$,

(2.) $(A_{SIG_{OS}}, belf_{OS}) \models f$ gilt für *kein* Tupel $(A_{SIG_{OS}}, belf_{OS})$,

(3.) $(A_{SIG_{OS}}, belf_{OS}) \models R_j(t_1,...,t_n) \quad \Leftrightarrow \quad (IT_{k_1}(t_1),...,IT_{k_n}(t_n)) \in r_j$,

(4.) $(A_{SIG_{OS}}, belf_{OS}) \models \neg F \quad \Leftrightarrow \quad (A_{SIG_{OS}}, belf_{OS}) \not\models F$,

(5.) $(A_{SIG_{OS}}, belf_{OS}) \models F_1 \wedge F_2 \quad \Leftrightarrow \quad (A_{SIG_{OS}}, belf_{OS}) \models F_1$
und $(A_{SIG_{OS}}, belf_{OS}) \models F_2$,

(6.) $(A_{SIG_{OS}}, belf_{OS}) \models F_1 \vee F_2 \quad \Leftrightarrow \quad (A_{SIG_{OS}}, belf_{OS}) \models F_1$
oder $(A_{SIG_{OS}}, belf_{OS}) \models F_2$,

(7.) $(A_{SIG_{OS}}, belf_{OS}) \models F_1 \underline{\vee} F_2 \quad \Leftrightarrow$ entweder $(A_{SIG_{OS}}, belf_{OS}) \models F_1$
oder $(A_{SIG_{OS}}, belf_{OS}) \models F_2$,

(8.) $(A_{SIG_{OS}}, belf_{OS}) \models F_1 \rightarrow F_2 \quad \Leftrightarrow \quad (A_{SIG_{OS}}, belf_{OS}) \not\models F_1)$
oder $(A_{SIG_{OS}}, belf_{OS}) \models F_2$,

(9.) $(A_{SIG_{OS}}, belf_{OS}) \models F_1 \leftrightarrow F_2 \quad \Leftrightarrow \quad (A_{SIG_{OS}}, belf_{OS}) \models F_1 \rightarrow F_2$
und $(A_{SIG_{OS}}, belf_{OS}) \models F_2 \rightarrow F_1$,

(10.) $(A_{SIG_{OS}}, belf_{OS}) \models \forall x: F \quad \Leftrightarrow \quad (A_{SIG_{OS}}, belf_{OS}[x/ob_u]) \models F$
für alle $ob_u \in OB_k$,

(11.) $(A_{SIG_{OS}}, belf_{OS}) \models \exists x: F \quad \Leftrightarrow \quad (A_{SIG_{OS}}, belf_{OS}[x/ob_u]) \models F$
für mindestens ein $ob_u \in OB_k$ und

(12.) $(A_{SIG_{OS}}, belf_{OS}) \models \underline{\exists} x: F \quad \Leftrightarrow \quad (A_{SIG_{OS}}, belf_{OS}[x/ob_u]) \models F$
für genau ein $ob_u \in OB_k$.

Eine ontologische Formel $F \in FORM_{SIG_{OS}}$ wird als *gültig* in der SIG_{OS}-Struktur $A_{SIG_{OS}}$ bezeichnet, wenn F durch *jede* Familie $belf_{OS}$ konzeptspezifischer Variablenbelegungen in $A_{SIG_{OS}}$ bestätigt wird. Verkürzt wird hierfür

$$A_{SIG_{OS}} \vDash F$$

geschrieben. Entsprechend wird die SIG_{OS}-Struktur $A_{SIG_{OS}}$ als *Modell* der ontologischen Formel F bezeichnet.

Die Vorgehensweise wird analog auf Formelmengen ausgeweitet. Demnach wird eine Formelmenge $FM \subseteq FORM_{SIG_{OS}}$ mit $FM=\{F_1,...,F_x\}$ durch eine Familie $belf_{OS}$ von konzeptspezifischen Variablenbelegungsfunktionen in einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ bestätigt, wenn die Formel $F_1 \wedge ... \wedge F_x$ durch $belf_{OS}$ in SIG_{OS} bestätigt wird. Hierfür wird die folgende Schreibweise verwendet:

$$(A_{SIG_{OS}}, belf_{OS}) \vDash FM$$
$$\text{mit } FM \subseteq FORM_{SIG_{OS}}, \text{ und } FM=\{F_1,...,F_x\} \text{ genau dann, wenn gilt:}$$
$$(A_{SIG_{OS}}, belf_{OS}) \vDash F_1 \wedge ... \wedge F_x.$$

Die Menge $FM \subseteq FORM_{SIG_{OS}}$ ontologischer Formeln ist genau dann in einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ *gültig*, wenn die Formel $F_1 \wedge ... \wedge F_x$ in $A_{SIG_{OS}}$ gültig ist:

$$A_{SIG_{OS}} \vDash FM$$
$$\text{mit } FM \subseteq FORM_{SIG_{OS}} \text{ und } FM=\{F_1,...,F_x\} \text{ genau dann, wenn gilt:}$$
$$A_{SIG_{OS}} \vDash F_1 \wedge ... \wedge F_x.$$

Die SIG_{OS}-Struktur $A_{SIG_{OS}}$ wird in diesem Fall als Modell der Menge FM ontologischer Formeln bezeichnet.

Mit der Funktion MOD_{OS} kann die Menge aller SIG_{OS}-Strukturen bestimmt werden, in denen eine ontologische Formel gültig ist. Die Menge MOD_{OS} zu einer ontologischen Formel $F \in FORM_{SIG_{OS}}$ umfasst alle SIG_{OS}-Strukturen, die Modelle von F sind:

$$MOD_{OS}: FORM_{SIG_{OS}} \rightarrow pot(A(SIG_{OS}))$$
$$\text{mit } MOD_{OS}(F) = \{A_{SIG_{OS}} \mid A_{SIG_{OS}} \vDash F\}.$$

Die Menge $MOD_{OS}(FM)$ aller Modelle zu einer Formelmenge $FM \subseteq FORM_{SIG_{OS}}$ mit $FM=\{F_1,...,F_x\}$ entspricht:

$$MOD_{OS}(FM) = \{A_{SIG_{OS}} \mid (A_{SIG_{OS}} \vDash (F_1 \wedge ... \wedge F_x))\}$$
$$\text{und somit } MOD_{OS}(FM)=MOD_{SS}(F_1 \wedge ... \wedge F_x).$$

Die Begriffe der *Erfüllbarkeit*, *Allgemeingültigkeit* und *Widersprüchlichkeit* können – analog zu Formeln über konventionellen und sortierten Signaturen – zur Charakterisierung von Formeln über ontologischen Signaturen beibehalten werden. Eine ontologische Formel $F \in FORM_{SIG_{OS}}$ ist genau dann erfüllbar, wenn es mindestens eine SIG_{OS}-Struktur $A_{SIG_{OS}}$ und mindestens eine Familie $belf_{OS}$ konzeptspezifischer Variablenbelegungen gibt, so dass F in $A_{SIG_{OS}}$ durch $belf_{OS}$ bestätigt wird. Eine Formelmenge $FM \subseteq FORM_{SIG_{OS}}$ wird entsprechend genau dann als *erfüllbar* bezeichnet, wenn es mindestens eine SIG_{OS}-Struktur $A_{SIG_{OS}}$ gibt, in der die Konjunktion $F_1 \wedge ... \wedge F_X$ aller ontologischen Formeln $F \in FM$ von mindestens einer Familie $belf_{OS}$ konzeptspezifischer Variablenbelegungen bestätigt wird.

Eine ontologische Formel $F \in FORM_{SIGOS}$ ist genau dann *allgemeingültig* oder *tautologisch*, wenn sie in jeder SIG_{OS}-Struktur gültig ist. Die Menge $MOD_{OS}(F)$ aller Modelle einer tautologischen ontologischen Formel $F \in FORM_{SIG_{OS}}$ stimmt mit der Menge $A(SIG_{OS})$ aller $SIG_{OS_{OS}}$-Strukturen zu der ontologischen Signatur SIG_{OS} überein. Eine Formelmenge FM ist genau dann tautologisch, wenn die Konjunktion $F_1 \wedge ... \wedge F_X$ aller ontologischen Formeln $F_x \in FM$ mit x=1,...,X allgemeingültig ist.

Eine ontologische Formel $F \in FORM_{SIG_{OS}}$ ist genau dann *widersprüchlich* oder *kontradiktorisch*, wenn keine SIG_{OS}-Struktur $A_{SIG_{OS}}$ und keine Familie $belf_{OS}$ konzeptspezifischer Variablenbelegungsfunktionen existieren, so dass F durch $belf_{OS}$ in $A_{SIG_{OS}}$ bestätigt wird. Die Modellmenge $MOD_{OS}(F)$ einer widersprüchlichen ontologischen Formel F ist somit leer. Darüber hinaus ist eine Menge $FM=\{F_1,...,F_x\}$ ontologischer Formeln genau dann widersprüchlich, wenn die konjunktive Verknüpfung $F_1 \wedge ... \wedge F_x$ aller Formeln aus FM widersprüchlich ist. Zu beachten ist hierbei, dass eine Formelmenge $FM \subseteq FORM_{SIG_{OS}}$ bereits dann widersprüchlich ist, wenn es mindestens eine ontologische Formel $F \in FM$ gibt, die widersprüchlich ist. Unabhängig von der Erfüllbarkeit der restlichen ontologischen Formeln $FM \backslash \{F\}$ kann es nämlich keine SIG_{OS}-Struktur $A_{SIG_{OS}}$ geben, in der die Formelmenge FM gültig ist, da die ontologische Formel F alle in Frage kommenden SIG_{OS}-Strukturen aufgrund ihrer Widersprüchlichkeit ausschließt.

Da der Begriff der Widersprüchlichkeit in einem späteren Kontext zur Kennzeichnung zulässiger Formeln gebraucht wird, wird die Menge

$$KOD_{SIG_{OS}} \subset FORM_{SIG_{OS}}$$

eingeführt. Die Menge $KOD_{SIG_{OS}}$ umfasst alle widersprüchlichen Formeln über einer ontologischen Signatur SIG_{OS}. Sie ist eine *echte* Teilmenge der Menge $FORM_{SIG_{OS}}$ aller ontologischen Formeln über der Signatur SIG_{OS}, da in $FORM_{SIG_{OS}}$ u.a. tautologische Formeln – wie z.B. die Formel w – enthalten sind, die in jeder SIG_{OS}-Struktur gültig sind.

Die Menge

$$KODM_{SIG_{OS}} \subset pot(FORM_{SIG_{OS}})$$

umfasst hingegen alle Mengen von Formeln über einer ontologischen Signatur SIG_{OS}, die widersprüchlich sind. Somit gilt für jedes Element $FM \in KODM_{SIG_{OS}}$ mit $FM=\{F_1,...,F_x\}$, dass keine SIG_{OS}-Struktur $A_{SIG_{OS}}$ und keine Familie $belf_{OS}$ konzeptspezifischer Variablenbelegungen existieren, so dass die Formel $F_1 \wedge ... \wedge F_x$ durch $belf_{OS}$ in $A_{SIG_{OS}}$ bestätigt wird.

Die Auswertung ontologischer Formeln unterscheidet sich formal nicht von der Auswertung konventioneller oder sortierter Formeln. Sämtliche Auswertungsverfahren sind über der induktiven Grammatik zur Konstruktion der jeweiligen Formelart definiert.

Ein wesentlicher Unterschied liegt in der Eigenart ontologischer Formeln, in ihren Argumenten auch mengenwertige Terme erfassen zu können. Dabei handelt es sich um solche Terme, deren extensionale Interpretation eine *Menge* formaler Objekte ist.[1)] Jedem Term $t \in TERM_k$ mit $k \in K_{MW}$ wird durch eine konzeptspezifische Termauswertungsfunktion IT_k statt eines einwertigen formalen Objektes, das nicht weiter in „elementare“ Bestandteile unterteilt werden kann (Skalar), ein solches formales Objekt $ob \in OB_k$ zugeordnet, das selbst eine Menge ist. In der Mengenfamilie OBF_{OS} aus einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ zu einer ontologischen Signatur SIG_{OS} sind nämlich auch solche konzeptspezifischen Objektmengen $OB_{k_x},...,OB_{k_y} \in OBF_{OS}$ enthalten, die Mengen von Skalaren umfassen. Entsprechend können in einer Relation $r \in RF$ aus einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ auch solche Objekttupel $(ob_1,...,ob_n)$ enthalten sein, bei denen die formalen Objekte $ob_1,...,ob_n$ Mengen sind.

Mit den Ausdrucksmitteln einer ontologischen Signatur SIG_{OS} können z.B. auch atomare ontologische Formeln der Art $F=R_j(t_1,...,t_x,...,t_n)$ konstruiert werden, bei denen nur[2)] der Term $t_x \in MT_k$ mengenwertig ist. Solche atomaren Formeln sind genau dann in einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ gültig, wenn sie durch jede Familie $belf_{OS}$ konzeptspezifischer Variablenbelegungen in $A_{SIG_{OS}}$ bestätigt werden. Damit die atomare Formel F in einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ gültig sein kann, muss in der SIG_{OS}-Struktur $A_{SIG_{OS}}$

$$(ob_1,...,ob_x,...,ob_n) \in r_j$$
$$\text{mit } IT_{k_i}(t_i)=ob_i$$
$$\text{für alle } i=1,...,x,...,n \text{ mit } x,n \in \mathbb{N}_+ \text{ und } 1 \leq x \leq n$$

gelten. Da die Auswertung mengenwertiger Terme auch immer zu mengenwertigen Individuen führt, ist auch das Individuum $IT_{k_x}(t_x)=ob_x$ mengenwertig. Es kann beispielsweise gelten $ob_x=\{ob_{x_1},...,ob_{x_m}\}$. Das Tupel $(ob_1,...,ob_x,...,ob_n)$ wäre in diesem Fall äquivalent zu dem Tupel $(ob_1,...,\{ob_{x_1},...,ob_{x_m}\},...,ob_n)$. Die atomare ontologische Formel $R_j(t_1,...,t_x,...,t_n)$ ist also in diesem Fall gültig in einer SIG_{OS}-Struktur $A_{SIG_{OS}}$, wenn in der Relation r_j das Objekttupel

$$(ob_1,...,\{ob_{x_1},...,ob_{x_m}\},...,ob_n)$$

enthalten ist.

Da mengenwertige Terme weder für konventionelle noch für sortierte Signaturen zulässig sind, können mit deren Ausdrucksmitteln auch keine Formeln mit mengenwertigen Termen im Argument konstruiert werden. Entsprechend dem Ausschluss solcher Formeln in konventionellen und sortierten Signaturen, braucht weder bei der Auswertung von konventionellen noch bei der Auswertung sortierter Formeln ein solcher Fall berücksichtigt zu werden. Im Fall ontologischer Formeln kann es hingegen durchaus sein, dass Formeln mit mengenwertigen Termen im Argument ausgewertet werden müssen. Für solche Konstruktionen sind die o.a. Besonderheiten zu berücksichtigen.

---

1) Vgl. Abschnitt 3.1.3.1.2.1.

2) Aus Gründen der Einfachheit wird lediglich der Fall betrachtet, bei dem nur der Term t_x und somit auch das formale Objekt $IT_{k.x}(t_x)=ob_x$ mengenwertig sind. Von den restlichen Termen im Argument der atomaren Formel $R_j(t_1,...,t_x,...,t_n)$ sei angenommen, dass sie einwertig sind.

3.1.3.2 Aspekte der intensionalen Semantik ontologischer Signaturen

3.1.3.2.1 Objektsprachliche Ausdrucksmittel

3.1.3.2.1.1 Konzepte

Als objektsprachliche Ausdrucksmittel sind in einer ontologischen Signatur SIG_{OS} Konzepte, Operationssymbole und Relationssymbole enthalten. Neben ihrer extensionalen Interpretation können den objektsprachlichen Ausdrucksmitteln aus SIG_{OS} auch intensionale Interpretationen zugeordnet werden. Dies ist insbesondere für *Konzepte* von Interesse, da sie das zentrale Modellierungsprimitiv in ontologischen Signaturen sind. Konzepte können in zweifacher Weise charakterisiert werden.

In ihrer *formalen* Charakterisierung werden Konzepte zunächst dazu verwendet, die Operations- und Relationssymbole aus einer ontologischen Signatur SIG_{OS} zu typisieren. Diese Typisierungen werden bei der Konstruktion von Termen und Formeln über SIG_{OS} berücksichtigt. Hinsichtlich dieses Punktes entsprechen Konzepte den *Sorten* aus sortierten Signaturen.

Sowohl Sorten als auch Konzepte werden darüber hinaus dazu verwendet, die signaturspezifischen Term- und Objektfamilien zu strukturieren. Sowohl die Mitglieder der Term- als auch die Mitglieder der Objektfamilien werden nämlich Sorten oder Konzepten zugeordnet. Dadurch heben sich beide Konstrukte von der konventionellen Prädikatenlogik ab. In der konventionellen Prädikatenlogik ist weder die Menge $TERM_{SIG_{KS}}$ aller Terme über einer konventionellen Signatur SIG_{KS} noch das prädikatenlogische Universum OB strukturiert.

Schließlich sind sowohl für Sorten als auch für Konzepte jeweils *Extensionen* vorgesehen, mit denen ihre *formale* Semantik festgelegt wird. Sorten werden durch die Interpretationsfunktion I_S auf sortenspezifische Objektmengen abgebildet. Die jeweilige Objektmenge OB_s entspricht der Extension der Sorte s. Konzepte werden durch die Interpretationsfunktion I_K auf konzeptspezifische Objektmengen OB_k abgebildet. Entsprechend werden die konzeptspezifischen Objektmengen als die Extensionen der jeweiligen Konzepte bezeichnet.

Aus formaler Sichtweise können somit zwischen Sorten und Konzepten keine wesentlichen Unterschiede ausgemacht werden. Dennoch wird als Unterschied zwischen Sorten und Konzepten aufgeführt, dass Erstgenannte nicht in den Argumenten von metasprachlichen Konstrukten vorkommen dürfen.[1] Dadurch seien beispielsweise metasprachliche Strukturierungen der Sortenmenge S im Gegensatz zur Konzeptmenge K nicht möglich. Einer solchen Unterscheidung kann allerdings aus zwei Gründen nicht gefolgt werden. Erstens ist die metasprachliche Strukturierbarkeit keine Eigenart von Konzepten selbst,

---

1) Vgl. MAIER (1993), S. 37 ff.

sondern eine Eigenart der metasprachlichen Strukturierungsrelationen. Somit kann nicht exklusiv für Konzepte der Anspruch der Strukturierbarkeit geltend gemacht werden. Zweitens werden auch im Rahmen der geordneten sortierten Prädikatenlogik (*order sorted logic*) binäre Relationen definiert, mit denen Unter- und Überordnungsbeziehungen zwischen Sorten ausgedrückt werden können.[1] Zwar wird die Strukturierung in der Regel lediglich auf eine metasprachliche Relation eingeschränkt, die in ihren wesentlichen Eigenschaften der Subkonzeptrelation $\sqsubseteq$ entspricht, allerdings existieren auch Ansätze, in denen darüber hinausgehende metasprachliche Strukturierungen vorgenommen werden können.[2]

Um die Unterschiede zwischen Sorten und Konzepten in ihren *materialen* Charakterisierungen zu beleuchten, muss die oftmals homonyme Verwendung der Bezeichnung *Konzept* aufgelöst werden. Als Konzepte kommen *mentale* Konzepte einerseits und *sprachliche* Konzepte andererseits in Frage. Bei mentalen Konzepten handelt es sich um Denkeinheiten, mit denen Akteure ihre realweltlichen Wahrnehmungen oder Vorstellungen strukturieren.[3] Sprachliche Konzepte sind hingegen formalsprachliche Bezeichner, mit denen mentale Konzepte referenziert werden können. In der bisherigen Verwendung der Bezeichnung *Konzept* wurden mentale Konzepte nicht thematisiert, sondern nur sprachliche Konzepte berücksichtigt. Auch im weiteren Verlauf wird stets auf sprachliche Konzepte Bezug genommen, wenn die Zusätze *sprachlich* und *mental* fehlen.

Durch den Umstand, dass es sich bei sprachlichen Konzepten um sprachliche Konstrukte handelt, wird nicht ausgeschlossen, dass es sich bei mentalen Konzepten auch um sprachliche Konstrukte handeln kann. Würde der Ausschluss sprachlicher Konstrukte als Denkeinheiten zugelassen, müsste angenommen werden, dass sprachunabhängiges Denken möglich sei. Diese Sichtweise wird allerdings seit längerem überwiegend abgelehnt[4]. Zudem wurde in dem wissenschaftstheoretischen Rahmen der vorliegenden Arbeit eine dazu entgegengesetzte Position eingenommen.[5] Somit wird für die vorliegen-

---

1) Vgl. BEIERLE ET AL. (1993), S. 179 ff.; GOGUEN/MESEGUER (1992), S. 217 ff.; HATZILYGEROUDIS/REICHGELT (1997), S. 256 ff.; MÜLLER (1999), S. 11 ff.; SCHMIDT-SCHAUß (1989), S. 13 ff.; SMOLKA (1989), S. 53 ff.

2) Vgl. KANEIWA (2001), S. 25 ff.; KANEIWA (2004), S. 158 ff.

3) Vgl. BUDIN (1996), S. 43; LÖBNER (2003), S. 257; SOWA (1984), S. 39.

4) Von den jüngeren Ausarbeitungen wird diese Position – die auch als *linguistic turn* bekannt geworden ist– insbesondere von WHORF vertreten; vgl. WHORF (2003), S. 11 ff. Eine abgeschwächte Form des *linguistischen Determinismus* nach WHORF findet sich beim CHOMSKY-Schüler PINKER unter der Bezeichnung *linguistische Relativität*; vgl. PINKER (1996), S. 67 ff. Während der erstgenannten Position zufolge die Denkkategorien durch Sprache erschlossen werden, wird nach der zweitgenannten Position lediglich die Ansicht vertreten, dass Unterschiede in Sprachen auch Unterschiede in den Gedanken der Sprecher bewirken. Für die vorliegende Arbeit wird von der erstgenannten Position ausgegangen.

5) Vgl. Abschnitt 2.1.1.

de Arbeit davon ausgegangen, dass Denken stets *in* einer Sprache stattfindet und somit durch die Ausdrucksmöglichkeiten einer Sprache potenzielle Denkinhalte bestimmt werden. Sprache wird somit nicht nur darauf reduziert, Instrument der *Kommunikation* von Gedanken zu sein. Vielmehr ist Sprache entsprechend dieser Sichtweise ein Instrument der *Konstruktion* von Gedanken. Unbestimmt bleibt allerdings hierbei, in *welcher* Sprache und in welcher *Art* von Sprache die mentale Konstruktionsleistung erfolgt. Der Verfasser sieht sich der Position verpflichtet, eine natürlichsprachliche Konzeptualisierung anzunehmen. Dadurch wird allen Positionen widersprochen, denen zufolge das Ergebnis eines Konzeptualisierungsprozesses ein formalsprachliches Konstrukt ist. Würde nämlich die Zulässigkeit formalsprachlicher Konzeptualisierungen erlaubt werden, so könnte der Prozess der Konstruktion einer Ontologie auf die triviale Übersetzung von einem formalsprachlichen Begriffssystem in ein anderes formalsprachliches Begriffssystem reduziert werden.[1)]

Mentale Konzepte werden somit in einer *internen Repräsentationssprache* des Akteurs ausgedrückt.[2)] Mit Hilfe seiner mentalen Konzepte ist es einem Akteur möglich, ein wiederum *mentales* oder *inneres Modell* einer realen oder gedachten Situation zu konstruieren.[3)] Dieses mentale Modell des Akteurs ist allerdings für Kommunikationsprozesse mit anderen Akteuren nicht geeignet, da es Ergebnis einer introvertierten Kognitionsleistung des Akteurs ist.[4)] Erst mit sprachlichen Konzepten ist es dem Akteur möglich, seine mentalen Konzepte anderen Akteuren mitzuteilen.

Sprachliche Konzepte sind mentalen Konzepten zugeordnet. Damit keine mehrdeutige Zuordnung von sprachlichen Konzepten durch die an der Ontologiekonstruktion und -verwendung beteiligten Akteure erfolgen kann, müssen sich die Akteure zuvor darüber geeinigt haben, welches sprachliche Konzept für welches mentale Konzept steht. Durch einen solchen Einigungsprozess wird eine eineindeutige Zuordnung konstruiert, die jedem sprachlichen Konzept k_n genau ein mentales Konzept zuordnet und umgekehrt. Sie entspricht der denotationalen Interpretation von sprachlichen Konzepten durch mentale Konzepte. Dieser Zusammenhang wird in der Abbildung 4 illustriert.

---

1) Vgl. ZELEWSKI (2005).

2) Vgl. HERRMANN ET AL. (1996), S. 121 ff.; JOHNSON-LAIRD (1983), S. 205 ff.; STACHOWIAK (1973), S. 207 ff.

3) Zur Konstruktion *mentaler* oder *innerer* Modelle vgl. JOHNSON-LAIRD (1983), S. 126 ff.; SCHMID/KINDSMÜLLER (1996), S. 285 f.; SCHÜTTE (1998), S. 60 ff.; SOWA (1984), S. 2, 27 u. 354; ZELEWSKI (1995), Bd. 2, S. 15 ff.

4) Vgl. ZELEWSKI (1995), Bd. 2, S. 15. Gleichwohl sind die mentalen Modelle Voraussetzung für die Kommunikation zwischen Akteuren; vgl. RAPAPORT (2003), S. 402; STACHOWIAK (1973), S. 208.

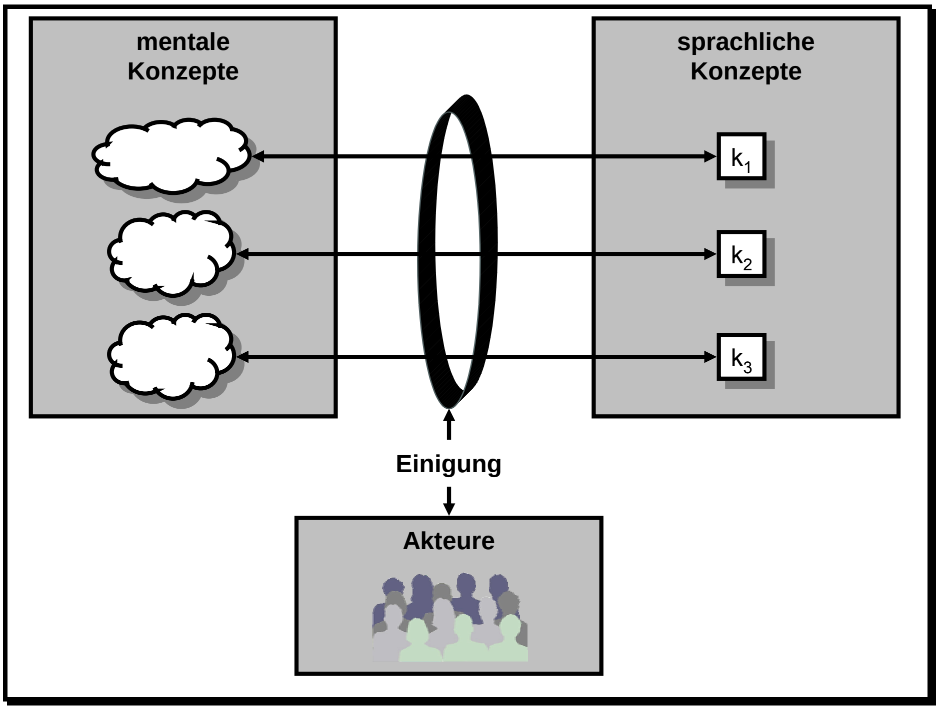

Abbildung 4: Ontologische Spezifikation mentaler Konzepte

Mit der Verwendung einer von mehreren Akteuren geteilten Ontologie wird stets auf einen mentalen Konzeptualisierungsausschnitt Bezug genommen, auf den sich alle beteiligten Akteure geeinigt haben. Ontologiegestützte Modelle korrespondieren mit den mentalen Modellen der beteiligten Akteure, da für die Konstruktion der Erstgenannten sprachliche Konzepte verwendet werden, die in einer eineindeutigen Beziehung zu den mentalen Konzepten stehen, mit denen die beteiligten Akteure ihre internen Modelle konstruieren.

Im Gegensatz zu mentalen Konzepten sind sprachliche Konzepte Konstrukte aus einem formalsprachlichen Alphabet $ALPH_{OS}$, über dem objektsprachliche Ausdrücke konstruiert werden können. Ihre denotationale Interpretation durch mentale Konzepte ist allerdings keine Eigenart der sprachlichen Konzepte, sondern Ergebnis des Einigungsprozesses der an der Ontologiekonstruktion und -verwendung beteiligten Akteure. Aus diesem Grund können auch aus materialer Sichtweise zwischen Sorten aus einer sortierten Signatur SIG_{SS} und sprachlichen Konzepten aus einer ontologischen Signatur SIG_{OS} keine Unterschiede ausgemacht werden, denn es kann nicht ausgeschlossen werden, dass auch Sorten aufgrund eines Einigungsprozesses der beteiligten Akteure mit mentalen Konzepten korrespondieren.

Aufgrund der sowohl formalen als auch materialen Ununterscheidbarkeit von Sorten und sprachlichen Konzepten werden daher im Folgenden die Bezeichnungen „Sorte“ und „sprachliches Konzept“ synonym verwendet. Trotz der Synonymie der beiden Bezeichner wird allerdings – ihren vorherrschenden Verwendungen folgend – bevorzugt, von „Sorten“ dann zu sprechen, wenn Strukturierungseinheiten aus einer sortierten Signatur SIG_{SS} gemeint sind. Die Bezeichnung „Konzept“ wird hingegen für die Strukturierungseinheiten aus einer ontologischen Signatur SIG_{OS} reserviert. Darüber hinaus wird im Folgenden – wie auch im bisherigen Verlauf – auf den Zusatz „sprachliche“ verzichtet. Wenn die Bezeichnung „Konzept“ ohne Zusatz verwendet wird, sind immer sprachliche Konzepte gemeint.

Die intensionale Semantik von Konzepten wird durch ihre Klassifikation näher bestimmt. Zu diesem Zweck erfolgen Klassifikationen bezüglich zweier Kriterien. Zum einen werden die Konzepte hinsichtlich ihrer Wertigkeit klassifiziert. Auf diese Klassifikationsmöglichkeit wurde durch die Unterscheidung zwischen den Konzeptmengen K_{EW} und K_{MW} hingewiesen. Zum anderen werden Konzepte mit Instanzen, die denotational durch reale Objekte interpretiert werden können, von Konzepten unterschieden, deren Instanzen selbstreferenziell sind. Formale Objekte die zu dieser letztgenannten Konzeptmenge gehören, sind beispielsweise Zeichenketten oder Zahlen. Sie werden von der Menge aller *Datenkonzepte* umfasst. Bei Konzepten, deren Instanzen reale Objekte repräsentieren können, handelt es sich um *Domänenkonzepte*. Während Domänenkonzepte dazu dienen, die Begrifflichkeiten aus der Domäne, die es zu modellieren gilt, zu repräsentieren, sind Datenkonzepte dazu da, jene Begrifflichkeiten zu spezifizieren, die zur Beschreibung *primitiver Datenmengen* benötigt werden.

Die Menge K_{DT} umfasst alle[1)] Datenkonzepte aus einer ontologischen Signatur SIG_{OS}. Die dazu komplementäre Menge $K \backslash K_{DT}$ umfasst alle Domänenkonzepte. Dabei ist nicht ausgeschlossen, dass die Menge $K \backslash K_{DT}$ der Domänenkonzepte leer ist. In diesem Fall beinhaltet die ontologische Signatur SIG_{OS} nur Datenkonzepte und die zugehörigen mengenwertigen Konzepte.

Die Menge K aller Konzepte setzt sich aus den zueinander disjunkten Mengen K_{EW} aller einwertigen Konzepte, der Menge K_{MW} aller mengenwertigen Konzepte und den beiden ausgezeichneten Konzepten $\top$ und $\bot$ zusammen:

$$K = K_{EW} \cup K_{MW} \cup \{\top, \bot\},$$
$$\text{mit } K_{EW} \cap K_{MW} = \varnothing$$
$$\text{und } K_{DT} \subseteq K_{EW}.$$

Während die einwertigen Konzepte aus der Menge K_{EW} als *originäre* Konzepte charakterisiert werden können, sind die mengenwertigen Konzepte aus der Menge K_{MW} *derivative* Konzepte. Sie werden von den *originären* Konzepten aus der Menge K_{EW} abge-

1) Bei der Konstruktion von Ontologien ist es üblich, nicht alle deskriptiven Symbole in einer Ontologie zu spezifizieren. Ontologien verfügen nämlich über die bemerkenswerte Eigenschaft der *Referenzierbarkeit*. Deskriptive Symbole, die benötigt werden und deren Spezifikationen aus einer öffentlich zugänglichen anderen Ontologie bekannt sind, können über ihre eindeutigen Identifikatoren wiederverwendet werden. Hierfür wird der Namensraum verwendet, der von XML-gestützten Ontologie-Sprachen zumeist unterstützt wird. Somit müssen nicht in jeder Ontologie die deskriptiven Symbole spezifiziert werden, die nicht von zentralem Interesse sind. Beispielsweise kann auf die reichhaltigen Spezifikationen aus *Ontolingua-Ontologien* referenziert werden, in denen u.a. auch Datenkonzepte spezifiziert sind; vgl. GRUBER (1993), S. 202 ff. Von diesen – für die Praxis zweifellos sehr nützlichen – Eigenarten von Ontologie-Sprachen wird allerdings im Folgenden abgesehen.

leitet.[1)] Die mengenwertigen Konzepte sind als Bilder der einwertigen Konzepte entsprechend der partiellen Mengenfunktion MEN bestimmt und sind somit derivativ konstruiert.

Die Extensionen der Konzepte aus der Menge K_{DT} sind in jeder SIG_{OS}-Struktur $A_{SIG_{OS}}$ identisch. In der Tabelle 5 ist ein Überblick über die Extensionen zu Konzepten aus K_{DT} und jeweils zugehörige natürlichsprachliche Erklärungen gegeben.

$k_n \in K_{DT}$	$I_K(k_n)=OB_{k_n}$	natürlichsprachliche Erklärung
Natural	$OB_{Natural}=\mathbb{N}$	Menge der natürlichen Zahlen
Integer	$OB_{Integer}=\mathbb{Z}$	Menge der ganzen Zahlen
$Integer_{pos}$	$OB_{Int_{pos}}=\mathbb{Z}_+=\mathbb{N}_+$	Menge der positiven ganzen Zahlen
$Integer_{neg}$	$OB_{Int_{neg}}=\mathbb{Z}_-$	Menge der negativen ganzen Zahlen
Real	$OB_{Real}=\mathbb{R}$	Menge der reellen Zahlen
$Real_{pos}$	$OB_{Real_{pos}}=\mathbb{R}_+$	Menge der positiven reellen Zahlen
$Real_{neg}$	$OB_{Real_{neg}}=\mathbb{R}_-$	Menge der negativen reellen Zahlen
Bool	$OB_{Bool}=\{true,false\}$	Menge der Wahrheitswerte
Char	$OB_{Char}=\{0,1,...,9,a,b,...,z,...\}$	Menge der Zeichen
String	$OB_{String}=OB_{Char}{}^*$	Menge der Zeichenketten

Tabelle 5: Extensionale Interpretation von Datenkonzepten

Konzepte aus K_{DT} werden mit Zahlenmengen, der Menge von Wahrheitswerten oder den Mengen der Zeichen und Zeichenketten extensional interpretiert. Die konzeptspezifischen Objektmengen, mit denen Datenkonzepte interpretiert werden, werden auch als *primitive Datenmengen* oder kurz *Datenmengen* bezeichnet.

Für die Bestimmung der Menge OB_{String} aller Zeichenketten wird die Menge OB_{Char} vorausgesetzt. Sie ist definiert als die Menge:

---

1) Vgl. PATIG (2001), S. 65.

$$OB_{Char}=\{0,1,...,9,a,b,...,z,A,B,...,Z,,,;,.,:,+,-,?,!,\$,\%;/,(,),=,\ ,\}.$$

Die Menge OB_{String} umfasst wiederum alle formalen Objekte, die durch Verkettung formaler Objekte aus der Menge OB_{Char} konstruiert werden können.[1] Dies wird durch den Kleene-Stern (*) im Definiens festgelegt. Daher sind auch alle Elemente der Menge OB_{Char} formale Objekte aus der konzeptspezifischen Objektmenge OB_{String}.[2)] Über die Objekte aus der Menge OB_{Char} hinaus beinhaltet die konzeptspezifische Objektmenge OB_{String} auch alle möglichen Verkettungen der formalen Objekte, die in OB_{Char} enthalten sind.

Die Menge OB_{Char}, die der Objektmenge OB_{String} zugrunde liegt, ist streng von der Menge $ALPH_{META}$ zu trennen, die eine Komponente jeder ontologischen Signatur SIG_{OS} ist. Während die Elemente der Menge OB_{Char} formale Objekte sind, die in ihrer atomaren oder zusammengesetzten Form zur extensionalen Interpretation von Termen zum Konzept *String* herangezogen werden können, sind die Elemente der Menge $ALPH_{META}$ rein metasprachliche Konstrukte. Die Elemente der Menge $ALPH_{META}$ werden in ihrer zusammengesetzten Form später zur metasprachlichen *Bezeichnung* und *Definition* von objektsprachlichen Konstrukten über einer ontologischen Signaturen SIG_{OS} verwendet. Würden für die Zwecke, bei denen das Alphabet $ALPH_{META}$ benötigt wird, die Objektmengen OB_{Char} und OB_{String} verwendet, würde das eine – zumindest partielle – extensionale Interpretation einer ontologischen Signatur SIG_{OS} voraussetzen. Demnach müssten für die Bezeichnung und Definition objektsprachlicher Konstrukte über SIG_{OS} formale Objekte aus einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ herangezogen werden. Auch wenn die Extensionen OB_{String} und OB_{Char} der Datenkonzepte *String* bzw. *Char* vollkommen invariant gegenüber Modellvariationen sind, läuft das dem Vorhaben zuwider, Ontologien ohne ihre extensionale Interpretation verwerten zu können. Daher wird das Alphabet $ALPH_{META}$ als eine Menge metasprachlicher Konstrukte *sui generis* betrachtet.

Sowohl bei ein- als auch bei mengenwertigen Konzepten handelt es sich um sprachliche Konzepte. Mengenwertige Konzepte unterscheiden sich von einwertigen Konzepten da-

1) Die Menge OB_{String} umfasst auch das leere Wort λ, da $OB_{Char}*$ als Monoid definiert ist. λ ist das neutrale Element aus $OB_{Char}*$.

2) Dies wird im weiteren Verlauf mit Hilfe der Subkonzeptrelation $\sqsubseteq$ festgelegt.

durch, dass sie in ihren Extensionen Mengen von Skalaren umfassen.[1)] Denn jede Instanz $ob_u \in OB_k$ eines mengenwertigen Konzepts $k \in K_{MW}$ entspricht selbst stets einer Menge. Dabei ist die Menge K_{MW} aller mengenwertigen Konzepte wie folgt definiert:

$$K_{MW} = \{MEN(k) \mid k \in K_{EW}\}$$
$$\text{mit } MEN: K_{EW} \rightarrow K_{MW}.$$

Um zu verdeutlichen, dass ein mengenwertiges Konzept MEN(k) von einem bestimmten einwertigen Konzept k abgeleitet ist, bieten sich verschiedene notationale Möglichkeiten an. In der ersten Darstellung kann ein mengenwertiges Konzept MEN(k) zu einem einwertigen Konzept $k \in K_{EW}$ entsprechend der Funktionsvorschrift als MEN(k) deklariert werden. Demnach wird für MEN(k) kein „eigenes" sprachliches Konzept erzeugt. Der Zugriff auf mengenwertige Konzepte erfolgt in diesem Fall nur über die Mengenfunktion MEN. Diese Darstellungsweise bietet sich an, wenn von Bedeutung ist, von welchem einwertigen Konzept k das mengenwertige Konzept MEN(k) abgeleitet ist.

Bei der zweiten Darstellungsweise werden mengenwertige Konzepte als eigenständige sprachliche Konzepte deklariert. Auf mengenwertige Konzepte ist bei dieser Alternative der Zugriff ohne Bezug auf das jeweils zugrunde liegende einwertige Konzept möglich, von dem sie abgleitet wurden. Diese Darstellungsweise bietet sich an, wenn das einwertige Konzept $k_1 \in K_{EW}$, aus dem das mengenwertige Konzept k_2 durch $MEN(k_1)=k_2$ hervorgeht, weniger von Relevanz ist. Beispielsweise kann für das mengenwertige Konzept MEN(Person), das aus dem einwertigen Konzept $Person \in K_{EW}$ abgeleitet ist, auch die Darstellungsweise „Personen" mit $Personen \in K_{MW}$ und Personen=MEN(Person) verwendet werden.

Die erste Alternative hat gegenüber der zweiten Alternative zum einen den Vorteil, ohne weitere Informationen auf die Wertigkeit eines beliebigen Konzepts schließen zu können. Aus der Schreibweise MEN(k) geht nämlich eindeutig hervor, dass es sich um

---

1) Für objektsprachliche Strukturierungseinheiten, die durch Mengen von Skalaren extensional interpretiert werden müssen, sind bislang unterschiedliche Verfahren bekannt. Beispielsweise werden im Rahmen der sortierten Prädikatenlogik Mengen*sorten* in sortierten Signaturen eingeführt, die durch *listenartige* Spezifikationen definiert werden. Diese Ausdrucksvariante kommt einer informationstechnischen Implementierung mit dem oft vorgegebenen Datentyp „Array" am nächsten. Dafür wird in einer sortierten Signatur zunächst die leere Liste „[]" als Konstantensymbol zu einer Mengensorte aufgeführt. Der Anschluss weiterer Terme an die Liste wird dann durch ein entsprechendes Operationssymbol umgesetzt; vgl. z.B. MÜLLER (1999), S. 20:

S:	{s,men_s};	
OPS:	[]	→ men_s
	anhaengen:	s men_s → men_s.

Die Listennotation wird teilweise auch zum Ausdrücken von *Multimengen* verwendet.

Zu Vorgehensweisen mit Ähnlichkeiten zu der hier vorgestellten Vorgehensweise vgl. ABITEBOUL/GRUMBACH (1991), S. 8 ff.; DOVIER ET AL. (2000), S. 870 ff.; KUPER (1987), S. 13 ff.; KUPER (1988), S. 10 ff.

ein mengenwertiges Konzept handeln muss.[1)] Zum anderen wird bei dem Versuch, die sprachlichen Konzepte an den natürlichen Sprachgebrauch anlehnend zu vergeben, die Gefahr einer mehrfachen Vergabe vermindert. Eine solche mehrfache Vergabe wäre nämlich bei der zweiten Alternative dann der Fall, wenn die Singular-Notation für Konzepte mit ihrer Plural-Notation übereinstimmt. Dies ist u.a. für solche Fälle von Relevanz, in denen es im natürlichen Sprachgebrauch üblich ist, das mengenwertige Konzept zu einem einwertigen Konzept mit dem gleichen Namen zu versehen. Beispielsweise würde das mengenwertige Konzept zum einwertigen Konzept *Mitarbeiter* auch *Mitarbeiter* lauten.

Bei der Funktion MEN handelt es sich um eine *partielle* Funktion. Es wird nämlich nicht von jedem einwertigen Konzept $k \in K_{EW}$ ein mengenwertiges Konzept abgeleitet. Es wird nur dann von einem einwertigen Konzept $k \in K_{EW}$ ein mengenwertiges Konzept $MEN(k) \in K_{MW}$ abgeleitet, wenn es notwendig ist, im Modell auch Mengen von Individuen aus einer konzeptspezifischen Objektmenge zu berücksichtigen. Daher ist die Mächtigkeit $|K_{MW}|$ der Menge K_{MW} maximal so groß wie die Mächtigkeit $|K_{EW}|$ der Menge K_{EW}.

Die Funktion MEN ist zudem *injektiv*. Jedes mengenwertige Konzept $MEN(k) \in K_{MW}$ ist nämlich das Bild von *höchstens* einem einwertigen Konzept $k \in K_{EW}$. Somit gilt:

$$\forall k_{n_1}, k_{n_2} \in K_{EW}, k_{n_3} \in K_{MW}: (MEN(k_{n_1})=k_{n_3} \wedge MEN(k_{n_2})=k_{n_3}) \rightarrow k_{n_1}=k_{n_2}.$$

Es handelt sich ebenso um eine *surjektive* Funktion. Die Menge K_{MW} stimmt nämlich mit der Bildmenge der Funktion MEN überein. Daher gilt:

$$\forall k_{n_2} \in K_{MW}: \exists k_{n_1} \in K_{EW}: MEN(k_{n_1})=k_{n_2}.$$

Somit handelt es sich bei MEN um eine partielle, *bijektive* Funktion.

Für die intensionale Beschreibung eines mengenwertigen Konzepts $MEN(k_n) \in K_{MW}$ wird auf die Intension des einwertigen Konzepts $k_n \in K_{EW}$ zurückgegriffen, von dem es abgeleitet ist. Entsprechend der Verzahnung der Intensionen ein- und mengenwertiger Konzepte sind ihre Extensionen voneinander abhängig. Die extensionale Interpretation eines mengenwertigen Konzept $MEN(k_n) \in K_{MW}$ erfolgt durch die konzeptspezifische Objektmenge:

$$OB_{MEN(k_n)}=\{\{ob_{1_1},...,ob_{1_{I(1)}}\},...,\{ob_{U_1},...,ob_{U_{I(U)}}\}\}.$$

Die Bestimmung der Mengen erfolgt über die Regel:

$$\forall k_n \in K_{EW}, MEN(k_n) \in K_{MW}: OB_{MEN(k_n)} = pot(OB_{k_n})$$
$$\text{mit } pot(OB_{k_n})=\{TOB \mid TOB \subseteq OB_{k_n}\}.$$

Jedem mengenwertigen Konzept $MEN(k) \in K_{MW}$ wird die Menge

---

1) Das setzt wiederum voraus, das jedes Konzept nur als eine Folge formalsprachlicher Buchstaben spezifiziert werden darf, bei der nur der erste Buchstabe ein Großbuchstabe ist. Hierauf wurde bereits in Fn. 1 auf S. 154 hingewiesen. Darüber hinaus wird vereinbart, die Zeichenfolge MEN zu reservieren. Dadurch kann die Zeichenfolge MEN(k) kein einwertiges Konzept sein.

$$OB_{MEN(k_n)}=pot(OB_{k_n})$$

zugeordnet, wobei $OB_{MEN(k_n)}$ die Potenzmenge der zugrunde gelegten Objektmenge OB_{k_n} ist.[1)] Somit sind in allen Mitgliedern einer Extension $OB_{MEN(k_n)}$ Individuen enthalten, die auch in der Extension OB_{k_n} enthalten sein müssen.

Das mengenwertige Konzept $MEN(k_n) \in K_{MW}$ muss bei dieser ersten Variante extensional durch die Potenzmenge $pot(OB_{k_n})$ der Extension OB_{k_n} des einwertigen Konzepts $k_n \in K_{EW}$ interpretiert werden. Wenn z.B. die Extension zu dem einwertigen Konzept $k_n \in K_{EW}$

$$OB_{k_n}=\{ob_1,ob_2,ob_3,ob_4\}$$

ist, dann ist die Extension $OB_{MEN(k_n)}$ des mengenwertigen Konzepts $MEN(k_n) \in K_{MW}$

$$\begin{aligned} OB_{MEN(k_n)}= \quad & \{\varnothing, \\ & \{ob_1,ob_2,ob_3,ob_4\}, \\ & \{ob_1,ob_2,ob_3\},\{ob_1,ob_2,ob_4\},\{ob_1,ob_3,ob_4\},\{ob_2,ob_3,ob_4\}, \\ & \{ob_1,ob_2\},\{ob_1,ob_3\},\{ob_2,ob_3\}, \\ & \{ob_1,ob_4\},\{ob_2,ob_4\},\{ob_3,ob_4\}, \\ & \{ob_1\},\{ob_2\},\{ob_3\},\{ob_4\}\}. \end{aligned}$$

Der Zusammenhang ist in der Abbildung 5 illustriert. Die gerichteten Kanten in der Abbildung repräsentieren jeweils die Teilmenge-von-Beziehung ($\subseteq$) zwischen einzelnen Teilmengen der Menge $OB_{MEN(k_n)}$

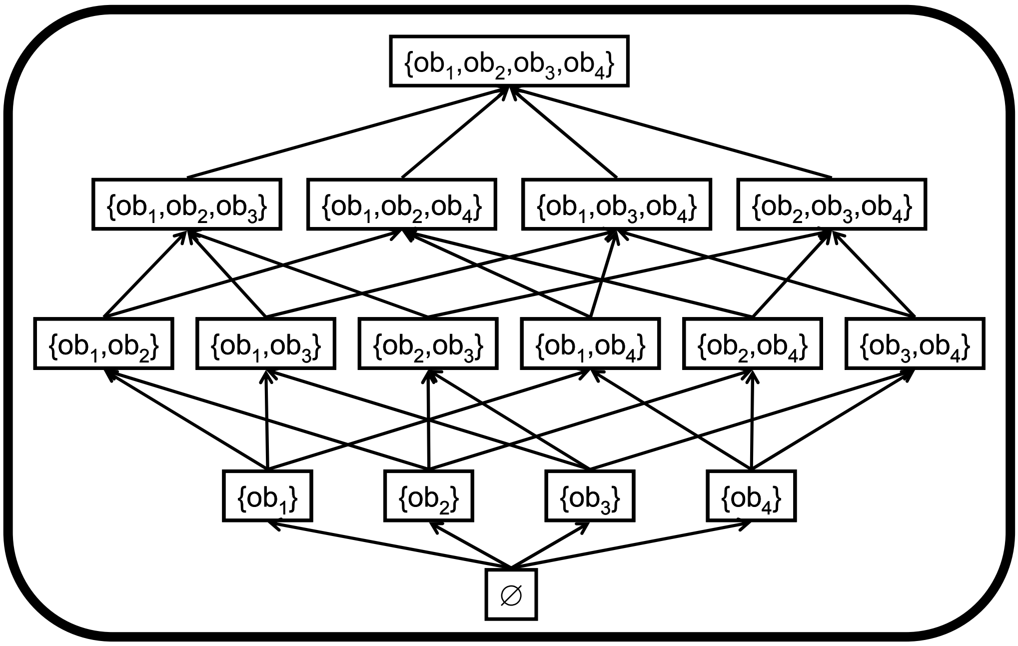

Abbildung 5: Konstruktion von Potenzmengen

1) Vgl. KUPER (1987), S. 13.

Zu beachten ist bei dieser Variante, dass die Mächtigkeit $|OB_{MEN(k_n)}|$ der Extension $OB_{MEN(k_n)}$ eines mengenwertigen Konzepts $MEN(k_n) \in K_{MW}$ im Verhältnis zur Mächtigkeit $|OB_{k_n}|$ der Extension OB_{k_n} des jeweils zugrunde liegenden einwertigen Konzepts $k_n \in K_{EW}$ immer exponentiell ansteigt. Wenn nämlich $|OB_{k_n}|=n$ gilt, dann gilt auch stets $|OB_{MEN(k)}|=2^n$. Bei einer „überschaubaren" Extension OB_{k_n} ist dies weniger problematisch. Weitaus schwieriger wird das hingegen, wenn beispielsweise die Extension OB_{k_n} 20 Instanzen umfasst. In diesem Fall müsste die Extension $OB_{MEN(k_n)}$ – inklusive der leeren Menge $\varnothing$ – bereits 1.048.576 Instanzen umfassen, wobei jede Instanz ein mengenwertiges Individuum wäre.

Um eine solche „explosionsartige" Vergrößerung der Mächtigkeit von Extensionen zu mengenwertigen Konzepten zu vermeiden, kann die oben angegebene Formel zur Bestimmung der Extension umformuliert werden zu:

$$\forall k_n \in K_{EW}, MEN(k_n) \in K_{MW}: OB_{MEN(k_n)} \subseteq pot(OB_{k_n})$$
$$\text{mit } pot(OB_{k_n}) = \{TOB \mid TOB \subseteq OB_{k_n}\}.$$

Demnach muss die Extension $OB_{MEN(k_n)}$ zu einem mengenwertigen Konzept $MEN(k_n)$ nicht mehr mit der Potenzmenge $pot(OB_{k_n})$ der Extension OB_{k_n} zu dem jeweils zugrunde liegenden einwertigen Konzept k_n *übereinstimmen*. Die Extension $OB_{MEN(k_n)}$ muss lediglich eine *Teilmenge* der Potenzmenge $pot(OB_{k_n})$ sein. Dadurch werden auch solche Mengen als Extensionen zu mengenwertigen Konzepten zugelassen, die nicht alle Teilmengen der Extension des einwertigen Konzepts als Mitglieder haben.

Allerdings ist bei dieser zweiten Variante die Extension $OB_{MEN(k)}$ nicht eindeutig bestimmt. Es ist nur festgelegt, dass sie eine Teilmenge der Potenzmenge von OB_k sein muss. Somit könnte die zweite Formulierung zur Bestimmung von mengenwertigen Extensionen lediglich als *Integritätsbedingung* verwendet werden. Eine SIG_{OS}-Struktur $A_{SIG_{OS}}$ wäre unzulässig bezüglich einer solchen Integritätsregel, wenn sie mindestens eine Extension $OB_{MEN(k_n)}$ zu einem mengenwertigen Konzept $MEN(k_n)$ umfassen würde, die keine Teilmenge der Menge $pot(OB_{k_n})$ wäre. Bei der ersten Variante konnte hingegen die Extension $OB_{MEN(k)}$ zu einem mengenwertigen Konzept $MEN(k)$ eindeutig von der Extension OB_k des jeweiligen einwertigen Konzepts k abgeleitet werden. In der folgenden Argumentation werden grundsätzlich beide Varianten zugelassen. Falls keine explizite Hervorhebung erfolgt, gilt stets die erste Variante zur Bestimmung der Extensionen mengenwertiger Konzepte.

Durch den Ausschluss von mengenwertigen Konzepten im Vorbereich der Funktion MEN können keine mengenwertigen Konzepte von wiederum mengenwertigen Konzepten abgeleitet werden. Es können nur mengenwertige Konzepte von einwertigen Konzepten abgeleitet werden. Würden nämlich auch solche mengenwertigen Konzepte zugelassen, die von wiederum mengenwertigen Konzepten abgeleitet wären, müssten die Extensionen der erstgenannten Mengen sein, deren Elemente selbst auch Mengen sind. Ein solcher rekursiver Aufruf der Funktion MEN wird dadurch vermieden, dass nur mengenwertige Konzepte zu einwertigen Konzepten zugelassen werden.

Die formalen Objekte aus den Extensionen ein- und mengenwertiger Konzepte werden auch als *ein-* oder *mengenwertige Individuen* bezeichnet. Ein einwertiges oder *skalares* Individuum ist demnach ein formales Objekt ob_u aus der Menge OB, das selbst keine Menge ist. Mengenwertige Individuen sind hingegen solche formalen Objekte aus OB, die Mengen sind. Sie umfassen stets nur einwertige Individuen als Elemente. Mit der Zulässigkeit mengenwertiger Individuen in ihrer formalen Semantik wird für Ontologien eine Ausdrucksmächtigkeit erschlossen, die gängigen Modellierungssprachen in der Regel verwehrt bleibt. Üblicherweise wird nämlich eine Begrenzung der Strukturierungshierarchie „nach unten" derart vorgenommen, dass Individuen hierarchisch nur durch die „(Nicht)-Instanz-von" -Beziehung zu Konzepten charakterisiert werden können.[1] Mit der Zulässigkeit mengenwertiger Individuen wird jedoch eine weitere Strukturierungsmöglichkeit erschlossen. Es handelt sich hierbei um die „(Nicht)-Element-von"-Beziehung, die zwischen einwertigen und mengenwertigen Individuen vorliegen kann. Mit Ontologien wird diese zusätzliche Strukturierungsmöglichkeit erschlossen, indem in der ontologischen Signatur SIG_{OS} Relationssymbole spezifiziert werden können, um die „Element von"- und die „Nicht-Element-von"-Beziehungen in Form ontologischer Formeln bzw. deren Negationen auszudrücken. Zudem können Operationssymbole spezifiziert werden, um beispielsweise die Vereinigung mengenwertiger Individuen in Form ontologischer Terme auszudrücken.[2]

Die Differenzierung von Individuen hinsichtlich ihrer Wertigkeit kann mit ihrer Differenzierung hinsichtlich ihrer Zusammengesetztheit zusammengeführt werden, wobei das Merkmal der Zusammengesetztheit vom Betrachtungswinkel abhängig ist, der eingeschlagen wird. Wird ein Individuum $ob \in OB_k$ aus einer k-spezifischen Objektmenge OB_k in seiner Eigenart als „originäres" Element von OB_k charakterisiert, handelt es sich um ein *einfaches Individuum*. Wird es hingegen in seiner Eigenart als Wert der Anwendung einer Operation $o_i \in OPF_{OS}$ auf ein n-Tupel $(ob_1,...,ob_n)$ formaler Objekte charakterisiert, handelt es sich um ein *zusammengesetztes Individuum*. Demnach kann ein einfaches Individuum $ob \in OB_k$ ebenso als zusammengesetztes Individuum $o_i(ob_1,...,ob_n)$ charakterisiert werden, wenn $ob=o_i(ob_1,...,ob_n)$ gilt. Ebenso können vermeintlich unterschiedliche Individuen $o_1(ob_{1_1},...,ob_{1_m})$ und $o_2(ob_{2_1},...,ob_{2_n})$ miteinander übereinstimmen, wenn die Anwendungen der Operationen o_1 und o_2 auf die jeweiligen m- bzw. n-Tupel von Individuen zum gleichen Individuum ob führen.

Aufgrund der binären Ausprägung beider Merkmale zur Klassifikation formaler Objekte können Variationen für das Aufkommen formaler Objekte in einer Vier-Felder-Tabelle dargestellt werden:

---

1) Vgl. KÜHNE/STEIMANN (2004), S. 116 f.

2) Die Spezifikation solcher Operations- und Relationssymbole wird in den Abschnitten 3.1.3.2.1.2 bzw. 3.1.3.2.1.3 exemplarisch aufgezeigt.

		Zusammengesetztheit	
		einfach	zusammengesetzt
Wertigkeit	einwertig	$ob \in OB_k$, mit $k \in K_{EW}$	$ob = o_i(ob_1,...,ob_n)$ mit $ob \in OB_k$ und $k \in K_{EW}$
	mengenwertig	$ob \in OB_k$, mit $k \in K_{MW}$ und $ob = \{ob_x,...,ob_y\}$	$ob = o_i(ob_1,...,ob_n) = \{ob_x,...,ob_y\}$ mit $ob \in OB_k$ und $k \in K_{MW}$

Tabelle 6: Klassifikation von Individuen aus SIG_{OS}-Strukturen

Es ist zu beachten, dass im Fall zusammengesetzter Individuen die Wertigkeit der Individuen im Argument der Operation o_i keine Rolle spielt. Es können beliebig viele der formalen Objekte $ob_1,...,ob_n$ im Argument der Operation o_i, die zur Konstruktion des komplexen formalen Objekts benötigt werden, mengenwertig sein und dennoch kann das formale Objekt ob selbst einwertig sein. Beispielsweise muss im Fall einer Operation

$$o_i: OB_{k_1} \times ... \times OB_{k_n} \rightarrow OB_k$$
$$\text{mit } k_1 \in K_{MW},\ k_a \in K_{EW} \text{ für } 2 \leq a \leq n \text{ und } k \in K_{EW}$$

der Wert $ob = o_i(ob_1,...,ob_n)$ mit $ob \in OB_k$ und $ob_u \in OB_{k_u}$ für alle $2 \leq u \leq n$ einwertig sein. Das formale Objekt ob_1 an der ersten Argumentstelle der Operation muss hingegen mengenwertig sein. Dies geht aus der Funktionsvorschrift für o_i hervor. Es kann beispielsweise $ob_1 = \{ob_x,...,ob_y\}$ und somit $o_i(\{ob_x,...,ob_y\},...,ob_n)$ gelten.

Die Konzepte $\top$ und $\perp$ gehören weder zu der Menge K_{EW} der einwertigen Konzepte noch zu der Menge K_{MW} der mengenwertigen Konzepte. $\top$ wird auch als *Maximalkonzept* und $\perp$ als *Minimalkonzept* bezeichnet. Teilweise werden das Maximal- bzw. Minimalkonzept als *Entität* oder *Ding* bzw. als *Absurdität* bezeichnet.[1)] Diese alternativen Bezeichnungen werden im weiteren Verlauf durch sprachspezifische Bezeichnungsfunktionen berücksichtigt.

Die extensionalen Interpretationen $OB_{\top}$ und $OB_{\perp}$ der Konzepte $\top$ und $\perp$ sind eindeutig bestimmt. Die Extension $OB_{\top}$ des Maximalkonzepts $\top$ umfasst alle Instanzen zu allen Konzepten in der ontologischen Signatur. Dies ist darauf zurückzuführen, dass alle Konzepte in der ontologischen Signatur in Subkonzeptrelation zum Maximalkonzept $\top$

1) Vgl. SOWA (2000), S. 54 bzw. 72.

stehen müssen. Auf diesen Aspekt der extensionalen Interpretation des Maximalkonzepts wird in Kürze vertieft eingegangen. Die Extension $OB_{\perp}$ des Minimalkonzepts $\perp$ ist hingegen stets die leere Menge $OB_{\perp}=\emptyset$. Demnach darf in der Extension $OB_{\perp}$ des Minimalkonzepts kein formales Objekt enthalten sein.

3.1.3.2.1.2 Operationssymbole

Operationssymbole werden in ontologischen Signaturen – im Gegensatz zur Vorgehensweise im Rahmen der konventionellen Prädikatenlogik – nicht mit einer natürlichen Zahl, sondern – analog zur Vorgehensweise im Rahmen der sortierten Prädikatenlogik – mit Konzeptfolgen typisiert.[1] Die Typisierung von Operationssymbolen ist bereits ein bedeutendes Ausdrucksmittel, um die Konstruktion ontologischer Ausdrücke zu untersagen, denen in keiner SIG_{OS}-Struktur $A_{SIG_{OS}}$ eine Extension zugewiesen werden kann, weil sie „sinnlos“ sind. Dieses Prinzip wurde bereits beim Übergang von der konventionellen zur sortierten Prädikatenlogik vorgestellt. Darüber hinaus sind mit Operationssymbolen in ontologischen Signaturen noch weitere Eigenschaften verbunden, um die intensionalen Aspekte von ontologischen Ausdrücken zu erfassen, in denen Operationssymbole verwendet werden. Hierzu gehört die Klassifikation von Operationssymbolen hinsichtlich ihrer *Wertigkeit*. Dadurch, das Operationssymbole hinsichtlich ihrer Wertigkeit klassifizieren lassen, kann bestimmt werden, welche Merkmale formale Objekte aus den Zielkonzepten von Operationssymbolen aufweisen müssen. Handelt es sich bei dem Zielkonzept k eines Operationssymbols um ein einwertiges Konzept, so muss aus der Anwendung der Operation o_i auf ein zulässiges Argument ein einwertiges Individuum – also ein Skalar – hervorgehen. Handelt es sich hingegen beim Zielkonzept k um ein mengenwertiges Konzept, so muss aus der Anwendung von o_i auf ein zulässiges Argument eine Menge von Skalaren hervorgehen.

Operationssymbole O_i, deren Zielkonzept $ZIEL_{OPS_{OS}}(O_i)=k_{n+1}$ mit $k_{n+1}\in K_{MW}$ ein mengenwertiges Konzept ist, das von einem einwertigen Konzept $k\in K_{EW}$ abgeleitet wurde, werden als *mengenwertige* und sonstige Operationssymbole als *einwertige*[2] Operationssymbole bezeichnet. Entsprechend kann die Menge OPS aller Operationssymbole in die Mengen OPS_{EW} aller *einwertigen Operationssymbole* und die Menge OPS_{MW} aller *mengenwertigen Operationssymbole* unterteilt werden:[3]

---

1) Auf die Unterschiede und Gemeinsamkeiten zwischen Sorten und Konzepten wurde bereits früher hingewiesen.

2) Der Zusatz *ontologisch* entfällt bei einwertigen und mengenwertigen Operationssymbolen, da es sich in jedem Fall um ontologische Operationssymbole handelt.

3) Zu beachten ist hierbei, dass die Unterteilung nur signaturspezifisch erfolgen kann. Die Menge OPS aller Operationssymbole ist nämlich unabhängig von einer bestimmten ontologischen Signatur definiert. Bei der Unterteilung der Menge OPS werden hingegen die Typisierungen von Operationssymbolen in Betracht gezogen, die signaturspezifisch sind. Im Grenzfall kann es vorkommen, dass ein Operationssymbol $O_i\in OPS$ bezüglich einer ersten ontologischen Signatur SIG_{OS_1} einwertig und bezüglich einer zweiten ontologischen Signatur SIG_{OS_2} mengenwertig ist.

$$OPS = OPS_{EW} \cup OPS_{MW}$$
$$\text{mit } OPS_{EW} \cap OPS_{MW} = \varnothing,$$
$$OPS_{EW} = \{O_i \mid O_i \in OPS \land ZIEL_{OPS_{OS}}(O_i) \in K_{EW}\}$$
$$\text{und } OPS_{MW} = \{O_i \mid O_i \in OPS \land ZIEL_{OPS_{OS}}(O_i) \in K_{MW}\}.$$

Die Extensionen einwertiger Operationssymbole sind solche Operationen, die n-Tupel von formalen Objekten auf jeweils ein einwertiges Individuum ob abbilden. Die Extension $o_i \in OPF$ mit o_i: $OB_{k_1} \times \ldots \times OB_{k_n} \rightarrow OB_{k_{n+1}}$ zu einem einwertigen Operationssymbol $O_i \in OPS_{EW}$ mit $typ_{OPS_{OS}} = (k_1 \ldots k_n, k_{n+1})$ mit $k_{n+1} \in K_{EW}$ hat in ihrem Nachbereich eine konzeptspezifische Objektmenge $OB_{k_{n+1}}$, deren Elemente einwertige Individuen sind. Mengenwertige Operationssymbole werden hingegen durch solche Operationen extensional interpretiert, deren Nachbereich Individuen umfasst, die selbst Mengen sind und somit mengenwertige Individuen darstellen. Die Extension $o_i \in OPF$ mit $o_i: OB_{k_1} \times \ldots \times OB_{k_n} \rightarrow OB_{k_{n+1}}$ zu einem mengenwertigen Operationssymbol $O_i \in OPS_{MW}$ hat nämlich in ihrem Nachbereich eine konzeptspezifische Menge $OB_{k_{n+1}}$, die eine *Potenzmenge* darstellt. Die Objektmenge $OB_{k_{n+1}}$ umfasst mit $k_{n+1} = MEN(k_x)$ und $OB_{k_{n+1}} = pot(OB_{k_x})$ solche Mengen als Elemente, die jeweils eine Teilmenge der Extension $OB(k_x)$ zu dem einwertigen Konzept k_x sind, von dem das mengenwertige Konzept k_{n+1} abgeleitet wurde.

Die konsequente Unterscheidung zwischen ein- und mengenwertigen Operationssymbolen spiegelt sich in einer bemerkenswerten Eigenart ontologischer Signaturen wider. Diese Eigenart ist bezüglich der ebenso konsequenten Unterscheidung zwischen ein- und mengenwertigen Termen einerseits und ein- und mengenwertigen Individuen andererseits vorhanden. So werden einwertige Operationssymbole unmittelbar zu einwertigen Termen, wenn es sich bei ihnen um Konstantensymbole handelt. Sie werden mittelbar zu einwertigen Termen, wenn es sich bei ihnen um mindestens einstellige Operationssymbole mit einem einwertigen Zielkonzept handelt und sie typgerecht auf ein Termtupel angewendet werden. Die Auswertung einwertiger Terme führt wiederum stets zu einwertigen Individuen, da die Termauswertung konzepttreu erfolgt. Analog hierzu gehen mengenwertige Terme u.a.[1)] aus mengenwertigen Operationssymbolen hervor und werden im Rahmen der konzepttreuen Termauswertung zu mengenwertigen Individuen ausgewertet.

Teilweise wird in der Literatur eine Einschränkung des Vorbereiches von Operationssymbolen auf die Menge $K \setminus K_{DT}$ aller Domänenkonzepte vorgenommen.[2)] Die Einschränkung erfolgt unter der Annahme, dass den Instanzen von Datenkonzepten in der SIG_{OS}-Struktur $A_{SIG_{OS}}$ keine Operationswerte zugeordnet werden sollen. Eine solche

1) Terme können auch Variablen sein, wodurch allerdings der o.a. Punkt nicht berührt wird.

2) Vgl. BENCH-CAPON/MALCOLM (1999), S. 252; BENCH-CAPON ET AL. (2003), S. 704.

Einschränkung findet sich auch in den Metamodellen[1)] von Sprachen wieder, die für die Konstruktion von Ontologien definiert wurden. Beispielsweise sind entsprechend dem Metamodell von RDF nur Aussagen (*Statements*) zugelassen, deren Vorbereich der Menge der *Resources*, deren Nachbereich aber der Vereinigungsmenge von Resources und *Literals* entspricht[2)]. Die Resources entsprechen der Menge $K \backslash K_{DT}$ aller Domänenkonzepte, während die Literals der Menge K_{DT} der Datenkonzepte entsprechen. Eine Unterscheidung zwischen ein- und mengenwertigen Operationssymbolen ist in RDF nicht vorgesehen.

Problematisch wird diese Einschränkung, wenn ein ontologiegestütztes Modell beispielsweise für die Menge der natürlichen Zahlen oder die Menge der Zeichenketten konstruiert werden soll. In einem solchen Modell könnte es von Interesse sein, Operationswerte für eine bestimmte Zahl oder eine bestimmte Zeichenkette zu definieren. Solche Konstruktionen sind zwar für das hier vorgestellte Vorhaben weniger von Relevanz, da es sich hier um einen Ansatz handelt, der primär die Modellierung *betrieblicher* Phänomene verfolgt. In betrieblichen Szenarien sind nämlich Eigenschaften von konkreten Zahlen oder Zeichenketten in der Regel nicht von Bedeutung. Jedoch sind auch Szenarien vorstellbar, in denen sie von Interesse sein können. Beispielsweise kann es sein, dass für Produkte Seriennummern als zeichenkettenartige Attribute definierbar sein müssen, bei denen die Zusammensetzung der Zeichenkette einem bestimmten Muster zu folgen hat. Darüber hinaus kann es auch sein, dass Attribute für Mengen von Individuen vergebbar sein sollen. Ein Beispiel hierfür sind Attribute von Aufträgen, die sich aus einer Menge von Produkten zusammensetzen. Wenn eine bestimmte Produktkonstellation gesondert ausgezeichnet werden muss, kann dies nur über Attribute erfolgen. Um keine „unnötigen“ Beschränkungen der Ausdrucksmächtigkeit des hier vorgestellten Ansatzes zu verantworten, werden im Vorbereich von Operationssymbolen alle Konzepte aus der Menge K zugelassen.

Eine ontologische Signatur SIG_{OS} sollte einerseits mindestens solche Operationssymbole umfassen, mit denen auf mengenwertige Terme zugegriffen werden kann. Für einen solchen Zugriff können Operationssymbole verwendet werden, mit denen die üblichen Symbole aus der Mengentheorie in einer ontologischer Signatur SIG_{OS} rekonstruiert werden. Solche Symbole werden einerseits als Operations- und andererseits als Relati-

1) Als *Metamodell* einer Sprache wird ein Modell der Modellierungsprimitive und ihrer Beziehungen bezeichnet, die eine Modellierungssprache zur Verfügung stellt; vgl. RAUH/STICKEL (1997), S. 101 ff.

2) Vgl. KLAPSING (2003), S. 41.

onssymbole eingeführt.[1)] Im ersten Fall werden Terme konstruiert, die selbst mengenwertig sind. Im zweiten Fall werden Aussagen über formale Objekte getätigt, die durch mengenwertige Terme repräsentiert werden. Zu den mengentheoretischen Symbolen zählen u.a. der Vereinigungsoperator $\cup$ und der Schnittmengenoperator $\cap$. Darüber hinaus kann es auch erforderlich sein, eine Operation zu beschreiben, mit der einem mengenwertigen Individuum seine Kardinalität zugeordnet werden kann.

Andererseits sollten in einer ontologischen Signatur SIG_{OS} Operationssymbole enthalten sein, mit denen Operationen auf Instanzen von Datenkonzepten beschrieben werden können. Typische Operationen auf Instanzen von Datenkonzepten sind die Addition, Subtraktion, Multiplikation und Division reeller Zahlen sowie die Konkatenation von Zeichen (und Zeichenketten) zu Zeichenketten. Operationssymbole zur Beschreibung von Operationen auf mengenwertigen Termen und auf Termen zu Datenkonzepten können wie folgt in einer ontologischen Signatur SIG_{OS} rekonstruiert werden:

1) Solche Symbole können entweder als Operations- oder als Relationssymbole eingeführt werden. Im ersten Fall werden neue Terme generiert, die selbst mengenwertig sind. Im zweiten Fall werden Aussagen über mengenwertige Individuen durch mengenwertige Terme getätigt. Grundsätzlich würde es ausreichen, die jeweiligen Symbole entweder nur als Operationssymbole oder nur als Relationssymbole einzuführen. Für Fälle, in denen in einer natürlichen Konzeptualisierung Relationssymbole angebracht wären, könnte als Zielkonzept des jeweiligen Operationssymbols das Datenkonzept *Bool* angegeben werden. Die Auflösung eventueller Redundanzen in den Ausdrucksmöglichkeiten würden dem jeweiligen Modellierungskontext überlassen werden. Zu einer ähnlichen Vorgehensweise vgl. ABITEBOUL/GRUMBACH (1991), S. 5.

O_i	$typ_{OPS_{OS}}$	$o_i=I_{OPS}(O_i)$
Intersection	k k, k mit $k \in K_{MW}$	intersection: $OB_k \times OB_k \rightarrow OB_k$ mit intersection(ob_1,ob_2)=$ob_1 \cap ob_2$
Union	k k, k mit $k \in K_{MW}$	union: $OB_k \times OB_k \rightarrow OB_k$ mit union(ob_1,ob_2) = $ob_1 \cup ob_2$
Card	k, Natural mit $k \in K_{MW}$	card: $OB_k \rightarrow OB_{Natural}$ mit card(ob) = $\|ob\|$
Add	Real Real, Real	add: $OB_{Real} \times OB_{Real} \rightarrow OB_{Real}$ mit add(ob_1,ob_2) = $ob_1 + ob_2$
Subt	Real Real, Real	subt: $OB_{Real} \times OB_{Real} \rightarrow OB_{Real}$ mit subt(ob_1,ob_2) = $ob_1 - ob_2$
Multip	Real Real, Real	multip: $OB_{Real} \times OB_{Real} \rightarrow OB_{Real}$ mit multip(ob_1,ob_2) = $ob_1 * ob_2$
Div	Real $Real_{POS}$, Real	div: $OB_{Real} \times OB_{Real_{POS}} \rightarrow OB_{Real}$ mit multip(ob_1,ob_2) = ob_1 / ob_2

Tabelle 7: Ausgezeichnete Operationssymbole in ontologischen Signaturen

Die Operationssymbole *Intersection* und *Union* sind in ihrem Zielbereich mit einem mengenwertigen Konzept $k \in K_{MW}$ und in ihrem Argumentbereich mit der Konzeptfolge $kk \in K^*$ typisiert. Das Operationssymbol *Card* ist hingegen in seinem Zielbereich mit dem Datenkonzept *Natural* und in seinem Argumentbereich mit einem Konzept k typisiert. Für das Konzept k bietet es sich bei den genannten Operationssymbolen an, die *obere Schranke* der Menge K_{MW} aller mengenwertigen Konzepte bezüglich der Subkonzeptrelation $\sqsubseteq$ zu verwenden.[1)] Dadurch wird gewährleistet, dass zur Konstruktion von Aussagen mit Hilfe der Operationssymbole *alle* mengenwertigen Terme an den Stellen eingesetzt werden können, für die k angegeben ist. Analog hierzu können an den Stellen, an denen die restlichen Operationssymbole mit *Real* typisiert sind, alle Terme eingesetzt werden, die numerische Individuen repräsentieren, da *Real* allen weiteren Datenkonzepten aus K_{DT} durch $\sqsubseteq$ übergeordnet ist, denen Zahlenmengen zugeordnet sind.[2)]

1) Die *obere Schranke* einer Konzeptmenge $KM \subseteq K$ bezüglich der Subkonzeptrelation $\sqsubseteq$ entspricht demjenigen Konzept $k \in K$, zu dem alle Konzepte $k_n \in KM$ in Subkonzeptrelation $\sqsubseteq$ stehen.

2) In Abschnitt 3.1.3.2.2.1 wird aufgezeigt, wie eine solche Ontologie konstruiert werden kann.

3.1.3.2.1.3 Relationssymbole

In seiner formalen Charakterisierung wird jedes Relationssymbol $R_j \in RS$ durch eine Relation $I_{RS}(R_j)=r_j$ aus einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ extensional interpretiert. Die Relation $r_j \in RF$ wird entsprechend als *Extension* des Relationssymbols R_j bezeichnet. In seiner materialen Charakterisierung entspricht ein Relationssymbol $R_j \in RS$ einer Beziehungsart, die zwischen realweltlichen Objekten bestehen kann.

Es empfiehlt sich, in eine ontologische Signatur SIG_{OS} solche Relationssymbole aufzunehmen, die für typische[1)] Aussagen über unterschiedliche Individuen benötigt werden. Beispielsweise kann ein Relationssymbol benötigt werden, um Aussagen über die Zugehörigkeit eines einwertigen Individuums ob_1 zu einem mengenwertigen Individuum ob_2 konstruieren zu können. Durch ein solches Relationssymbol wird die „Element-von"-Relation ($\in$) zwischen formalen Objekten ausgedrückt. Die Negation einer damit konstruierten Formel entspricht hingegen der „Nicht-Element-von"-Relation ($\notin$). Darüber hinaus können Relationssymbole benötigt werden, um Vergleichsrelationen (z.B. $\geq$, $>$) auszudrücken. Tabelle 8 gibt einen Überblick über solche Relationssymbole, ihre Typisierung in einer ontologischen Signatur SIG_{OS} und ihre Extensionen.

R_j	$typ_{RS_{OS}}(R_j)$	$I_{RS}(R_j)=r_j$
Equal	$\top\ \top$	$equal \subseteq OB_\top \times OB_\top$ mit $equal=\{(ob_1,ob_2) \mid ob_1=ob_2\}$
Element_of	$k_1\ k_2$ mit $k_1 \in K_{EW}, k_2 \in K_{MW}$ und $k_2=MEN(k_1)$	$element\_of \subseteq OB_{k_1} \times OB_{k_2}$ mit $element\_of=\{(ob_1,ob_2) \mid ob_1 \in ob_2\}$
Greater	Real Real	$greater \subseteq OB_{Real} \times OB_{Real}$ mit $greater=\{(ob_1,ob_2) \mid ob_1 > ob_2\}$
Greater_or_Equal	Real Real	$greater\_or\_equal \subseteq OB_{Real} \times OB_{Real}$ mit $greater\_or\_equal=\{(ob_1,ob_2) \mid ob_1 \geq ob_2\}$

Tabelle 8: Ausgezeichnete Relationssymbole in ontologischen Signaturen

Da das Relationssymbol *Element_of* für jede ontologische Signatur SIG_{OS} in Abhängigkeit von der oberen Schranke aus der Konzeptmenge K definiert werden muss, die durch die Subkonzeptrelation $\sqsubseteq$ konstruiert wird, ist die Typisierung von *Element_of* in der Tabelle 8 lediglich beispielhaft aufgezeigt. Die Relationssymbole *Equal*, *Greater* und *Greater_or_Equal* sind hingegen unabhängig von der jeweiligen ontologischen Signatur SIG_{OS} mit ($\top\ \top$) bzw. (Real Real) typisiert.

1) Solche Relationssymbole finden sich in der Regel als „Built-In"-Ausdrucksmittel in Werkzeugen zur Verarbeitung von Ontologien; vgl. ONTOPRISE (2003), S. 12 ff.

Mit dem Relationssymbol *Equal* können Aussagen bezüglich der Gleichheit zweier Individuen konstruiert werden. Dabei können im Argument von *Equal* Terme zu beliebigen Konzepten eingesetzt werden, da jeder ontologische Term auch in der Termmenge $TERM_T$ enthalten sein muss.

In den Argumenten der beiden Relationssymbole *Greater* und *Greater_Or_Equal* können hingegen nur Terme zum Konzept *Real* eingesetzt werden. Dabei sollten die Konzepte $Real_{pos}$, $Real_{neg}$, $Integer_{pos}$, $Integer_{neg}$ und *Integer* alle in Subkonzeptrelation $\sqsubseteq$ zum Konzept *Real* stehen, so dass alle Terme zu diesen Konzepten auch in der Argumenten von *Greater* und *Greater_or_Equal* eingesetzt werden können. Eine Konzeptstruktur, in der eine solche Ordnung von Konzepten durch die Subkonzeptrelation $\sqsubseteq$ definiert ist, wird im folgenden Abschnitt vorgestellt. In diesen Fällen umfasst die Menge $TERM_{Real}$ aller Terme zum Konzept *Real* alle Terme aus den konzeptspezifischen Termmengen $TERM_{Real_{pos}}$, $TERM_{Real_{neg}}$, $TERM_{Integer_{pos}}$, $TERM_{Integer_{neg}}$ und $TERM_{Integer}$. Entsprechend können in den Argumenten der erwähnten Relationssymbole alle Terme vorkommen, die numerischer Art sind.

3.1.3.2.2 Metasprachliche Ausdrucksmittel

3.1.3.2.2.1 Metasprachliche Strukturierungsrelationen

Metasprachliche Strukturierungsrelationen sind ein wesentlicher Baustein bei dem Vorhaben, die intensionale Semantik objektsprachlicher Komponenten von ontologischen Signaturen auszudrücken. Als Strukturierungsrelationen werden solche metasprachlichen Relationen verstanden, die dazu verwendet werden, Konzepte aus einer ontologischen Signatur SIG_{OS} in derartige Beziehungen zueinander zu setzen, aufgrund derer für die jeweiligen Konzept-Extensionen relationsspezifische Restriktionen zu berücksichtigen sind. So werden aufgrund der metasprachlichen Spezifikation von Strukturierungsrelationen denkbare Extensionen zu Konzepten intensional ausgeschlossen.

Unter die Strukturierungsrelationen fallen die Subkonzeptrelation $\sqsubseteq$, die Äquivalenzrelation $\triangleq$ und die Inkompatibilitätsrelation $\curlyvee$. Dabei dient die Subkonzeptrelation $\sqsubseteq$ der *vertikalen* Strukturierung der Konzeptmenge K. Durch sie werden nämlich Über- und Unterordnungsbeziehungen zwischen Konzepten ausgedrückt. Die beiden restlichen Strukturierungsrelationen dienen der *horizontalen* Strukturierung der Konzeptmenge K.

Auf der Konzeptmenge K ist durch die *Subkonzeptrelation* $\sqsubseteq$ eine *partielle Ordnung* definiert. Sie ist insofern partiell, als dass nicht für jedes Konzeptpaar $k_1,k_2 \in K$ die Beziehung $k_1 \sqsubseteq k_2$ oder die Beziehung $k_2 \sqsubseteq k_1$ gelten muss. Bei den Elementen der Subkonzeptrelation $\sqsubseteq$ handelt sich um Konzeptpaare der Form (k_1,k_2), bei denen die Objektmenge OB_{k_1}, die das erste Konzept k_1 extensional interpretiert, immer eine Teilmenge der Objektmenge OB_{k_2} ist, die das zweite Konzept k_2 extensional interpretiert. Die partielle Ordnung entspricht den folgenden Eigenschaften der *Subkonzeptrelation* $\sqsubseteq$:[1)]

1) Vgl. ERDMANN (2001), S. 77.

Reflexivität $\forall k \in K$: $k \sqsubseteq k$

Antisymmetrie $\forall k_1,k_2 \in K$: $(k_1 \sqsubseteq k_2 \wedge k_2 \sqsubseteq k_1) \rightarrow k_1 = k_2$

Transitivität $\forall k_1,k_2,k_3 \in K$: $(k_1 \sqsubseteq k_2 \wedge k_2 \sqsubseteq k_3) \rightarrow k_1 \sqsubseteq k_3$.

Die partielle Ordnung auf der Menge K durch die Subkonzeptrelation $\sqsubseteq$ wird durch das Tupel $(K,\sqsubseteq)$ ausgedrückt. Dabei wird das Tupel $(K,\sqsubseteq)$ auch verkürzt als *Taxonomie* und entsprechend die Subkonzeptrelation $\sqsubseteq$ als *taxonomische Relation* aufgefasst. Erstgenanntes kann als *Digraph*[1)] mit der Knotenmenge K und der Kantenmenge $\sqsubseteq$ aufgefasst werden. Jedes Element der Kantenmenge ist eine gerichtete Kante von einem Ursprungs- zu einem Zielknoten. Der Ursprungsknoten k_1 jeder Kante $(k_1 \sqsubseteq k_2)$ ist dem Zielknoten k_2 taxonomisch untergeordnet. Stehen zwei Konzepte $k_1,k_2 \in K$ in der Subkonzeptbeziehung der $k_1 \sqsubseteq k_2$ zueinander, wird k_1 als *Subkonzept* des Konzepts k_2 bezeichnet.[2)] Das Konzept k_2 ist dann ein *Super-* oder *Oberkonzept* des Konzepts k_1.[3)] Aufgrund der Transitivität der *Subkonzeptrelation* müssen zwei Konzepte k_1 und k_2 nicht in einem unmittelbaren[4)] Ordnungsverhältnis zueinander stehen.

---

1) Die Bezeichnung „Digraph“ ist eine Abkürzug für „directed Graph“. Als Digraph werden mathematische gerichtete Graphen bezeichnet.

2) Alternative Bezeichner für Subkonzept sind *Unterkonzept* und *Hyponym*; vgl. CARSTENSEN ET AL. (2001), S. 387 f.; LÖBNER (2003), S. 118.

3) Alternative Bezeichner für *Superkonzept* sind *Oberkonzept* und *Hyperonym*; vgl. CARSTENSEN ET AL. (2001), S. 387 f.; LÖBNER (2003), S. 118.

4) Die Relation $\sqsubseteq$ entspricht der Vereinigung der Relation $\sqsubseteq_d \subseteq K \times K$ mit ihrem transitiven und reflexiven Abschluss, wobei $\sqsubseteq_d$ als eine unmittelbare („direkte“) Ordnung zwischen je zwei Konzepten definiert ist; vgl. HATZILYGEROUDIS/REICHGELT (1997), S. 255, allerdings lediglich mit dem Verweis auf den transitiven Abschluss der unmittelbaren Konzeptordnung.

Der reflexive Abschluss $r(\sqsubseteq_d)$ der Relation $\sqsubseteq_d$ ist definiert als

$$r(\sqsubseteq_d) := \sqsubseteq_d \cup \{(k,k) | k \in K\}.$$

Der transitive Abschluss $t(\sqsubseteq_d)$ der Relation $\sqsubseteq_d$ ist definiert als

$$t(\sqsubseteq_d) := \cup_{n \in \mathbb{N}} u_n(\sqsubseteq_d), \text{ mit}$$
$$u_0(\sqsubseteq_d) := \sqsubseteq_d \text{ und}$$
$$u_{n+1}(\sqsubseteq_d) := \{(k_1,k_2) \in K \times K | \exists k \in K: (k_1,k) \in u_n(\sqsubseteq_d) \wedge (k,k_2) \in \sqsubseteq_d\} \setminus \cup_{r \leq n} u_r(\sqsubseteq_d) \text{ für alle } n \in \mathbb{N}.$$

Es gilt daher $\sqsubseteq := \sqsubseteq_d \cup r(\sqsubseteq_d) \cup t(\sqsubseteq_d)$; vgl. EHRIG ET AL. (1999), S. 81 f.

Die Unterscheidung zwischen einer mittelbaren und unmittelbaren Unterordnung von Konzepten kommt auch einer implementierungsnahen Spezifikation zu Gute. Transitive Relationen lassen sich in deklarativen Programmiersprachen als *rekursive* Regeln umsetzen. Rekursive Regeln sind dadurch gekennzeichnet, dass sie die Mengen der Formeln erweitern, indem sie auf bereits bestehende Formeln zurückgreifen; vgl. AMBLE (1987), S. 48. Bei einer Umsetzung einer Konzepthierarchie in der Programmiersprache PROLOG ist es notwendig, rekursive Regeln so zu definieren, dass sie „terminieren“. Die Termination einer rekursiven Regel in PROLOG ist dann gewährleistet, wenn die Regel eine „Abbruchbedingung“ beinhaltet. Bei einer Regel der Form

$$sub(s_1,s_3) :- sub(s_1,s_2), sub(s_2,s_3)$$

Taxonomische Strukturen $(K,\sqsubseteq)$, in denen mindestens ein Konzept k_1 existiert, das zwei weiteren Konzepten k_2 und k_3 unmittelbar durch $\sqsubseteq$ untergeordnet ist, werden als *netzartige Taxonomien* bezeichnet. Netzartige Taxonomien können somit als *heterarchische* Strukturen charakterisiert werden. Im Gegensatz hierzu wird in hierarchischen Strukturen die maximal *ein*fache Unterordnung von Elementen gegenüber anderen Elementen zugelassen. Eine Taxonomie $(K,\sqsubseteq)$, die einer hierarchischen Struktur entspricht, liegt somit genau dann vor, wenn jedes Konzept $k_n \in K$ höchstens einem weiteren Konzept durch $\sqsubseteq$ untergeordnet ist.

Die inverse Relation zu der Subkonzeptrelation $\sqsubseteq$ ist die *Superkonzeptrelation* $\sqsupseteq$. Für die Superkonzeptrelation $\sqsupseteq$ gilt:

$$\forall k_1,k_2 \in K: k_1 \sqsubseteq k_2 \leftrightarrow k_2 \sqsupseteq k_1.$$

Da für die Superkonzeptrelation $\sqsupseteq$ ebenfalls die Eigenschaften der Reflexivität, Antisymmetrie und Transitivität gelten, fundiert auch sie eine partielle Ordnung. Die partiellen Ordnungen von $\sqsubseteq$ und $\sqsupseteq$ werden als *zueinander duale* Ordnungen bezeichnet.[1] Dabei ist das Maximalkonzept $\top$ Superkonzept zu jedem Konzept $k \in K$ aus der einer ontologischen Signatur SIG_{OS}:

$$\forall k \in K: (k \sqsubseteq \top).$$

Das Minimalkonzept $\perp$ ist Subkonzept zu jedem Konzept $k \in K$ aus der ontologischen Signatur SIG_{OS}:

$$\forall k \in K: (\perp \sqsubseteq k).$$

Die Menge aller Super- bzw. Subkonzepte zu einem Konzept lassen sich mittels der Funktionen

$$\text{sup}: K \rightarrow pot_+(K)$$
$$\text{mit sup}(k)=\{k_n \mid (k \sqsubseteq k_n)\}$$

$$\text{bzw. sub}: K \rightarrow pot_+(K)$$
$$\text{mit sub}(k)=\{k_n \mid (k_n \sqsubseteq k)\}$$

bestimmen. Der Funktionswert sup(k) zu einem Konzept $k \in K$ ist die Menge von Konzepten, in der alle Superkonzepte von k enthalten sind. Es kann sich hierbei nicht um die leere Menge $\varnothing$ handeln, da das Maximalkonzept $\top$ in der Menge sup(k) zu jedem Konzept $k \in K$ enthalten ist. Der Funktionswert sub(k) umfasst entsprechend alle Subkonzepte von k. Auch hierbei ist die leere Menge $\varnothing$ als Bild der Funktion sub zu einem

---

ist dies nicht gegeben. In der o.a. Prolog-Formel entsprechen das Zeichen „:-“ und das Komma „,“ dem Subjugatspfeil „→“ bzw. dem Konnektor „∧“ der Prädikatenlogik. Der PROLOG-Interpreter würde bei der obigen Regel im Suchbaum eine *Endlos-Schleife* durchlaufen. Eine Lösungsmöglichkeit besteht darin, im Regel-Rumpf die unmittelbare Konzeptordnungsrelation „direct_sub“ aufzunehmen:

$$sub(s_1,s_3) \text{ :- } direct\_sub(s_1,s_2),sub(s_2,s_3).$$

1) Vgl. GANTER/WILLE (1996), S. 5.

Konzept k ausgeschlossen, da in der Menge sub(k) zu jedem Konzept k∈K mindestens das Minimalkonzept $\perp$ enthalten sein muss. Darüber hinaus ist wegen der Reflexivität der Subkonzeptrelation $\sqsubseteq$ das Konzept k sowohl in sup(k) als auch in sub(k) enthalten.

Die Subkonzeptrelation $\sqsubseteq$ kann sowohl aus dem Blickwinkel der *extensionalen* als auch aus dem Blickwinkel der *intensionalen* Semantik charakterisiert werden.[1)] Aus dem Blickwinkel der extensionalen Semantik wird durch die taxonomische Unterordnung eines sprachlichen Konzepts k_1 gegenüber einem zweiten sprachlichen Konzept k_2 eine Anforderung an die Beschaffenheit der konzeptspezifischen Objektmenge OB_{k_2} formuliert. Denn mit der Subkonzeptbeziehung $k_1 \sqsubseteq k_2$ wird ausgedrückt, dass die Extension OB_{k_1} in jeder SIG_{OS}-Struktur $A_{SIG_{OS}}$ eine Teilmenge der Extension von OB_{k_2} sein muss. Es gilt demnach:

$$\forall k_1,k_2 \in K: (k_1 \sqsubseteq k_2) \rightarrow (OB_{k_1} \subseteq OB_{k_2}).$$ [2)]

Daher gehört jedes Individuum ob_u, das zu der k_1-spezifischen Objektmenge OB_{k_1} gehört, auch zu der k_2-spezifischen Objektmenge OB_{k_2}, wenn $k_1 \in K$ ein Subkonzept von $k_2 \in K$ ist. Diese Anforderung an die Beschaffenheit konzeptspezifischer Objektmengen in SIG_{OS}-Strukturen wurde bereits in deren Definition berücksichtigt.[3)]

Aus intensionaler Perspektive wird die Semantik eines sprachlichen Konzepts k durch die Menge sup(k) aller Konzepte festgelegt, denen gegenüber es taxonomisch untergeordnet ist. Denn die intensionale Semantik eines Konzepts stimmt mit der Menge aller Merkmale überein, denen zufolge ein formales Objekt eine Instanz des Konzepts darstellt. Die intensionale Semantik eines Konzepts k kann in einer Ontologie dadurch präzisiert werden, dass es allen Konzepten taxonomisch untergeordnet wird, die den Merkmalen entsprechen, durch die k intensional beschrieben werden kann. Die Präzisierung der intensionalen Semantik sprachlicher Konstrukte wird auf den folgenden Seiten anhand einer so genannten *Kreuzklassifikation* beschrieben.

Dadurch, dass die Subkonzeptrelation $\sqsubseteq$ als partielle Ordnung auf der Konzeptmenge K definiert ist, ergibt sich für die konzeptspezifischen Term- und Objektmengen, dass sie entsprechend $\sqsubseteq$ untereinander in einer Teilmengenbeziehung stehen müssen. Dies resultiert aus der Transitivität der Subkonzeptrelation $\sqsubseteq$. In der Abbildung 6 ist das Prinzip der Teilmengenbeziehung zwischen konzeptspezifischen Term- und Objektmengen verdeutlicht. Die gerichteten Kanten auf der linken Seite verdeutlichen die Teilmengenbeziehungen zwischen konzeptspezifischen Termmengen. Analog dazu verdeutlichen die

---

1) Vgl. BERGAMASCHI/SARTORI (1992), S. 386 ff.

2) Vgl. (mit direktem Bezug zu Ontologien) BAADER ET AL. (2004), S. 10; BENSLIMANE ET EL. (2002), S. 61; ERDMANN (2001), S. 77; MÄDCHE ET AL. (2003), S. 289.
Vgl. (ohne direkten Bezug zu Ontologien) BEIERLE ET AL. (1993), S. 194; EHRICH ET AL. (1989), S. 178; GOGUEN/MESEGUER (1992), S. 226; HATZILYGEROUDIS/REICHGELT (1997), S. 256; KANEIWA (2001), S. 30; LAUSEN/VOSSEN (1996), S. 87 ff.; LOECKX ET AL. (1996), S. 235; MÜLLER (1999), S. 12.

3) vgl. Abschnitt 3.1.3.1.1

gerichteten Kanten auf der rechten Seite die Teilmengenbeziehungen zwischen konzeptspezifischen Objektmengen. Die gerichteten Kanten in der Mitte repräsentieren die taxonomische Unterordnung der aufgeführten Konzepte. Die gerichteten Kanten, die von den Seiten in die Mitte führen, verdeutlichen jeweils, welchen Konzepten die Term- bzw. Objektmengen zugeordnet sind.

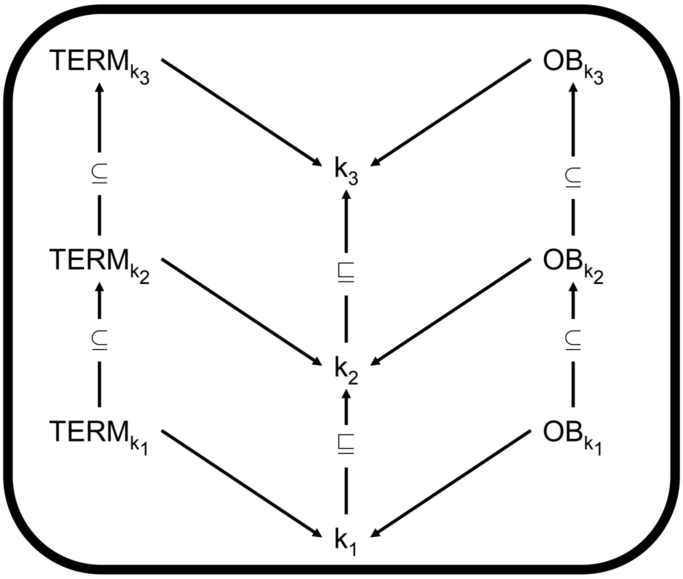

Abbildung 6: Teilmengenbeziehung zwischen konzeptspezifischen Objektmengen

Die konzeptspezifischen Termmengen $TERM_{k_1}$, $TERM_{k_2}$ und $TERM_{k_3}$ und die konzeptspezifischen Objektmengen OB_{k_1},OB_{k_2} und OB_{k_3} zu den Konzepten $k_1,k_2,k_3 \in K$ stehen in den Teilmengenbeziehungen

$$TERM_{k_1} \subseteq TERM_{k_2} \subseteq TERM_{k_3}$$

bzw.

$$OB_{k_1} \subseteq OB_{k_2} \subseteq OB_{k_3}$$

zueinander, wenn

$$k_1 \sqsubseteq k_2 \sqsubseteq k_3$$

gilt.

Da das Maximalkonzept $\top$ Superkonzept zu allen anderen Konzepten in der ontologischen Signatur SIG_{OS} ist, umfasst die Objektmenge $OB_{\top}$, mit der $\top$ extensional interpretiert wird, alle Individuen aus allen konzeptspezifischen Objektmengen:

$$\forall k \in K: OB_k \subseteq OB_{\top}.$$

Damit stimmt die konzeptspezifische Objektmenge $OB_{\top}$ mit der Menge OB überein, die als Vereinigung aller konzeptspezifischen Objektmengen vorgestellt wurde und dem prädikatenlogischen Universum entspricht:

$$OB_{\top} = OB.$$

Die Subkonzeptrelation $\sqsubseteq$ ist über dem kartesischen Produkt $(K \times K)$ definiert. Somit ist sie sowohl über der Menge K_{EW} der einwertigen als auch über der Menge K_{MW} der mengenwertigen Konzepte definiert. Dabei darf kein Konzept k_1 einem Konzept k_2 untergeordnet werden, wenn k_1 und k_2 unterschiedliche Wertigkeiten aufweisen. Ansonsten könnte beispielsweise die Extension eines einwertigen Konzepts $k_1 \in K_{EW}$ sowohl einwertige als auch mengenwertige Individuen umfassen, wenn ein weiteres allerdings mengenwertiges Konzept $k_2 \in K_{MW}$ gegenüber k_1 durch $k_2 \sqsubseteq k_1$ untergeordnet würde. Somit dürfen in den Mengen $sup(k_1)$ bzw. $sub(k_1)$ zu einem einwertigen Konzept $k_1 \in K_{EW}$ außerhalb des Maximal- und Minimalkonzepts $\top$ bzw. $\bot$ nur einwertige Konzepte vorkommen. Gewährleistet wird dies durch die metasprachlichen Integritätsregeln

$$\forall k_1,k_2 \in K: (k_2 \in sup(k_1) \wedge k_1 \in K_{EW} \wedge k_2 \neq \top) \rightarrow k_2 \in K_{EW}$$
$$\text{und } \forall k_1,k_2 \in K: (k_2 \in sub(k_1) \wedge k_1 \in K_{EW} \wedge k_2 \neq \bot) \rightarrow k_2 \in K_{EW}.$$

Analog dürfen in den Mengen $sup(MEN(k_1))$ bzw. $sub(MEN(k_1))$ zu einem mengenwertigen Konzept $MEN(k_1) \in K_{MW}$ außerhalb des Maximal- bzw. Minimalkonzepts nur mengenwertige Konzepte vorkommen:

$$\forall k_1,k_2 \in K: (k_2 \in sup(k_1) \wedge k_1 \in K_{MW} \wedge k_2 \neq \top) \rightarrow k_2 \in K_{MW}$$
$$\text{und } \forall k_1,k_2 \in K: (k_2 \in sub(k_1) \wedge k_1 \in K_{MW} \wedge k_2 \neq \bot) \rightarrow k_2 \in K_{MW}.$$

Zwecks Trennung der Menge K_{EW} aller einwertigen Konzepte von der Menge K_{MW} aller mengenwertigen Konzepte bietet es sich an, in einer ontologischen Signatur SIG_{OS} zwei *obere Schranken* für die beiden Konzeptmengen K_{EW} und K_{MW} zu spezifizieren. Es handelt sich dabei um ein einwertiges bzw. mengenwertiges Konzept, zu dem *alle* einwertigen bzw. mengenwertigen Konzepte in Subkonzeptrelation $\sqsubseteq$ stehen.[1)] In Abbildung 7 ist eine schematische Darstellung oberer Schranken wiedergegeben. Die gerichteten Kanten repräsentieren jeweils eine taxonomische Unterordnungsbeziehung.

1) *Untere* Schranken einer Konzeptmenge sind wiederum Subkonzepte aller anderen Konzepte aus der betrachteten Konzeptmenge. Zu Schranken in partiellen Ordnungen vgl. GANTER/WILLE (1996), S. 5.

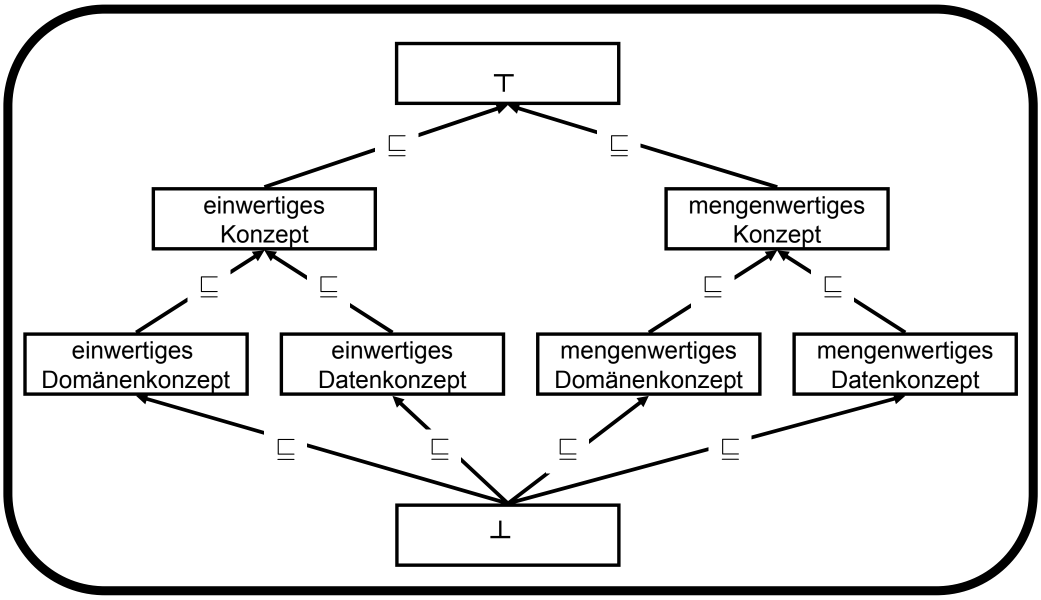

Abbildung 7: Oberste und unterste Schranken

Das Konzept *einwertiges_Konzept* ist in Abbildung 7 die obere Schranke der Menge K_{EW}. Alle einwertigen Domänen- und Datenkonzepte stehen in Subkonzeptrelation $\sqsubseteq$ zu *einwertiges_Konzept*. Das Konzept *mengenwertiges_Konzept* ist analog die obere Schranke der Menge K_{MW}. Sowohl alle mengenwertigen Domänenkonzepte als auch alle mengenwertigen Datenkonzepte stehen in Subkonzeptrelation $\sqsubseteq$ zu *mengenwertiges_Konzept*.

Eine taxonomische Ordnung auf der Menge K_{DT} aller Datenkonzepte kann entsprechend Abbildung 8 spezifiziert werden.[1)]

1) Vgl. z.B. MÜLLER (1999), S. 13.

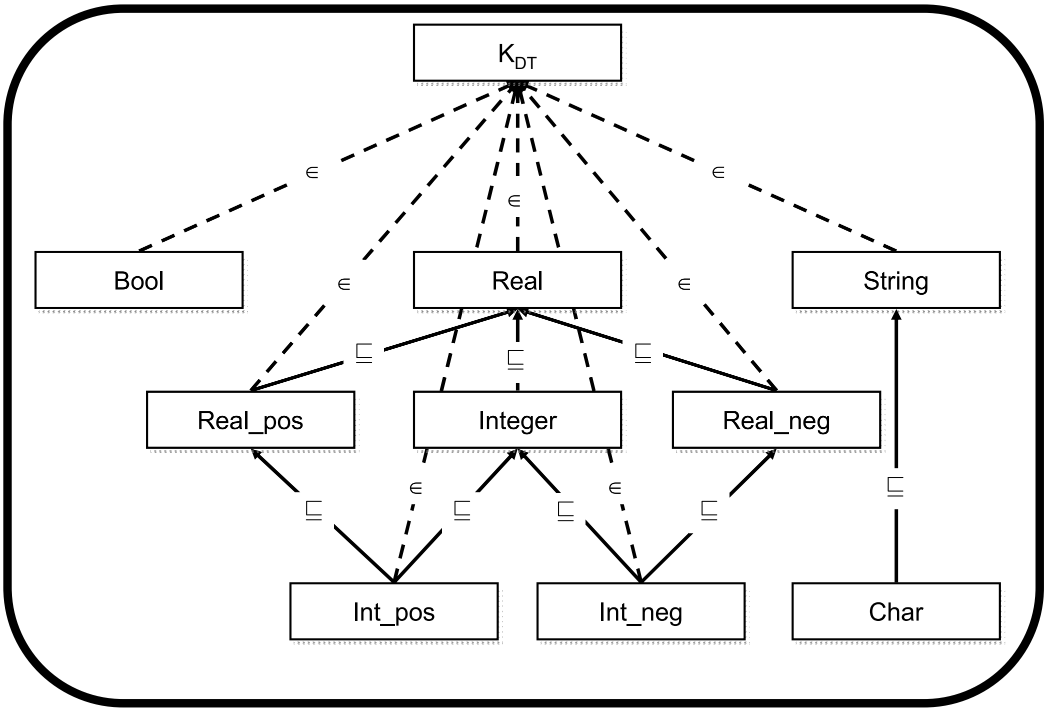

Abbildung 8: Ordnung auf der Menge K_{DT} der Datenkonzepte

Das Konzept *Real* ist die obere Schranke der Menge aller Datenkonzepte, mit denen numerische Werte beschrieben werden können. So wird z.B. das Konzept *Integer*, das extensional durch die Menge $OB_{Integer}=\mathbb{Z}$ der ganzen Zahlen interpretiert wird, dem Konzept *Real* untergeordnet, das wiederum durch die Menge $OB_{Real}=\mathbb{R}$ der reellen Zahlen interpretiert wird.

Eine Taxonomie auf der Menge K_{MW} wird seltener angesprochen.[1)] Während die eine Teilmenge $\{(k_1,k_2) \mid k_1,k_2 \in K_{EW}\}$ der Relation $\sqsubseteq$ einer konventionellen Taxonomie einwertiger Konzepte entspricht, ist die dazu komplementäre Teilmenge $\{(k_3,k_4) \mid k_3,k_4 \in K_{MW}\}$ komplexer. Dabei steht die Partial-Taxonomie auf der Menge K_{MW} in gegenseitigem Abhängigkeitsverhältnis mit der Partial-Taxonomie auf der Menge K_{EW}. Für zwei mengenwertige Konzepte $MEN(k_1),MEN(k_2) \in K_{MW}$ mit $k_1,k_2 \in K_{EW}$ gilt nämlich, dass sie genau dann in einer Subkonzeptrelation $\sqsubseteq$ stehen, wenn die einwertigen Konzepte $k_1,k_2 \in K_{EW}$, aus denen sie abgeleitet werden, in Subkonzeptrelation stehen und zu jedem einwertigen Konzept $k \in K_{EW}$ auch ein mengenwertiges Konzept $MEN(k) \in K_{MW}$ definiert ist:

$$\forall k_1,k_2 \in K_{EW}: (k_1 \sqsubseteq k_2) \leftrightarrow (MEN(k_1) \sqsubseteq MEN(k_2)).$$

Da MEN als partielle Funktion definiert ist, kann es jedoch sein, dass zu einem einwertigen Konzept $k \in K_{EW}$ kein mengenwertiges Konzept spezifiziert ist. Für diesen Fall gilt die „abgeschwächte” Regel:

1) Vgl. MÜLLER (1999), S. 49.

$$\forall k_3,k_4 \in K_{MW}\ \exists k_1,k_2 \in K_{EW}:$$
$$k_3=MEN(k_1) \wedge k_4=MEN(k_2) \wedge k_3 \sqsubseteq k_4 \rightarrow k_1 \sqsubseteq k_2.$$

Durch die beiden Regeln wird gewährleistet, dass die Partial-Taxonomien $(K_{EW},\sqsubseteq)$ und $(K_{MW},\sqsubseteq)$ nicht zueinander widersprüchlich sind. Entsprechend müssen die Extensionen zweier mengenwertiger Konzepte $MEN(k_1)$ und $MEN(k_2)$ in Teilmengenbeziehung $OB_{MEN(k_1)} \subseteq OB_{MEN(k_2)}$ zueinander stehen, wenn die zugehörigen einwertigen Konzepte in der Subkonzeptbeziehung $k_1 \sqsubseteq k_2$ zueinander stehen.[1] Wenn beispielsweise $OB_{MEN(k_1)}=pot(OB_{k_1})$ und $OB_{MEN(k_2)}=pot(OB_{k_2})$ gelten, dann ist die konzeptspezifische Objektmenge $OB_{MEN(k_1)}$ eine Teilmenge der konzeptspezifischen Objektmenge $OB_{MEN(k_2)}$.[2] Der Zusammenhang ist in der Abbildung 9 dargestellt:

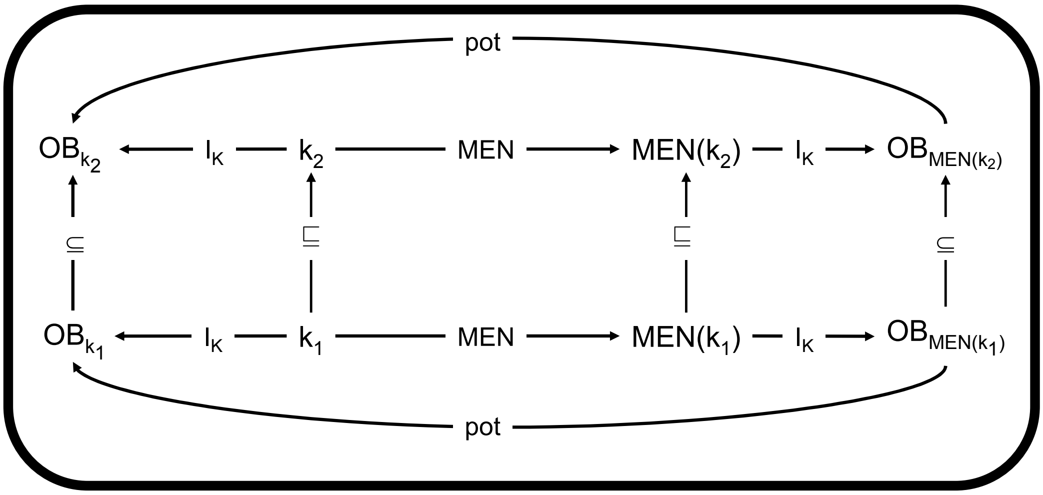

Abbildung 9: Extensionen ein- und mengenwertiger Konzepte

Wenn z.B. das Konzept k_1 durch die Objektmenge $OB_{k_1}=\{ob_1,ob_2\}$ und das Konzept k_2 mit $k_1 \sqsubseteq k_2$ durch die Objektmenge $OB_{k_2}=\{ob_1,ob_2,ob_3\}$ extensional interpretiert werden, dann können die mengenwertigen Konzepte $MEN(k_1)$ und $MEN(k_2)$ durch

---

1) Dies gilt allerdings nur, wenn die Extensionen der mengenwertigen Konzepte jeweils mit den Potenzmengen der Extensionen der jeweiligen einwertigen Konzepte übereinstimmen. Aufgrund der explosionsartigen Vergrößerung der Mächtigkeit von Extensionen zu mengenwertigen Konzepten wurde allerdings zuvor auch erlaubt, dass sie nur eine *Teilmenge* der Potenzmenge sind (vgl. Abschnitt 3.1.3.2.1.1). In diesem Fall gilt die Notwendigkeit zur Inklusion mengenwertiger Konzepte nicht.

2) Der Beweis für den Zusammenhang ergibt sich aus der mengentheoretischen Axiomatisierung. Wenn

(1.) $OB_{k_1} \subseteq OB_{k_2}$ wegen $k_1 \sqsubseteq k_2$,

(2.) $OB_{MEN(k_1)}=pot(OB_{k_1})=\{TOB \mid TOB \subseteq OB_{k_1}\}$ und

(3.) $OB_{MEN(k_2)}=pot(OB_{k_2})=\{TOB \mid TOB \subseteq OB_{k_2}\}$ gelten, dann gilt für jede Menge

(4.) $TOB \in pot(OB_{k_1})$ aufgrund (1.)

(5.) $TOB \subseteq OB_{k_2}$ und somit $TOB \in pot(OB_{k_2})$.

$OB_{MEN(k_1)}=\{\varnothing,\{ob_1\},\{ob_2\},\{ob_1,ob_2\}\}$ bzw.
$OB_{MEN(k_2)}=\{\varnothing,\{ob_1\},\{ob_2\},\{ob_3\},\{ob_1,ob_2\},\{ob_1,ob_3\},\{ob_2,ob_3\},\{ob_1,ob_2,ob_3\}\}$

mit $OB_{MEN(k_1)}\subseteq OB_{MEN(k_2)}$ extensional interpretiert werden.

Sollen heterarchische Strukturen in taxonomischen Ordnungen ausgeschlossen werden, so müssen zwei Konzepte k_2 und k_3 in Subkonzeptrelation $\sqsubseteq$ oder Superkonzeptrelation $\sqsupseteq$ zueinander stehen, wenn sie jeweils als Superkonzepte zu einem Konzept k_1 festgelegt sind.[1)] Für die Subkonzeptrelation $\sqsubseteq$ ist der Ausschluss heterarchischer Strukturen allerdings nicht notwendig. Es kann zur natürlichen Repräsentation eines Realitätsausschnitts sogar notwendig sein, ein Konzept k_1 als Subkonzept zu zwei unterschiedlichen Konzepten k_2 und k_3 festzulegen, die weder in Beziehung $k_2\sqsubseteq k_3$ noch in der Beziehung $k_3\sqsubseteq k_2$ zueinander stehen. Dieses Phänomen wird als *Kreuzklassifikation*[2)] eines Konzepts bezeichnet und umfasst solche Fälle, bei denen die konzeptspezifische Objektmenge OB_{k_1} zu dem untergeordneten Konzept k_1 jeweils eine Teilmenge von zwei konzeptspezifischen Objektmengen OB_{k_2} und OB_{k_3} zu zwei Konzepten k_2 und k_3, die untereinander in keiner Unterordnungsbeziehung stehen, ist. Es gelten also bei einer Kreuzklassifikation:

$$k_2 \sqsupseteq k_1 \sqsubseteq k_3,\ \neg(k_2\sqsubseteq k_3 \vee k_3\sqsubseteq k_2)$$
$$\text{und } OB_{k_2} \supseteq OB_{k_1} \subseteq OB_{k_3}.$$

gelten. Wenn die konzeptspezifische Objektmenge OB_{k_1} nicht leer ist, ist auch die Schnittmenge $OB_{k_2}\cap OB_{k_3}$ nicht leer. Wenn die konzeptspezifische Objektmenge OB_{k_1} nicht leer ist, müssen die beiden konzeptspezifischen Objektmengen OB_{k_2} und OB_{k_3} in ihrer Schnittmenge $OB_{k_2}\cap OB_{k_3}$ mindestens die Individuen enthalten, die auch in OB_{k_1} enthalten sind:

$$\forall k_1,k_2,k_3\in K: (k_1 \sqsubseteq k_2 \wedge k_1 \sqsubseteq k_3) \rightarrow (OB_{k_1} \subseteq (OB_{k_2} \cap OB_{k_3})).$$

Darüber hinaus können allerdings in der Schnittmenge $OB_{k_2}\cap OB_{k_3}$ auch solche Individuen enthalten sein, die nicht in OB_{k_1} enthalten sind. Dies ist genau dann der Fall, wenn ein formales Objekt ob_u sowohl zu der Extension OB_{k_2} des Konzepts k_2 als auch zu der Extension OB_{k_3} des Konzepts k_3 gehört, allerdings nicht zu der Extension OB_{k_1} des Konzepts k_1.

Die Kreuzklassifikation kann hilfreich sein, um die intensionale Semantik von Konzepten mit Hilfe der Subkonzeptrelation $\sqsubseteq$ zu bestimmen. Aus diesem Blickwinkel müsste ein Konzept k allen Konzepten untergeordnet werden, die jenen Merkmalen zugeordnet sind, die auf k zutreffen. Beispielsweise kann die Intension des Konzepts *Frau* durch die Merkmale „weibliches Wesen" und „Mensch" angegeben werden. Entsprechend müsste das Konzept *Frau* den beiden Konzepten *Weibliches_Wesen* und *Mensch* taxonomisch untergeordnet werden.

---

1) Vgl. LUTZEIER (1995), S. 75. Demnach müsste gelten:

$$\forall k_1,k_2,k_3\in K: ((k_1 \sqsubseteq k_2) \wedge (k_1 \sqsubseteq k_3)) \rightarrow ((k_2 \sqsubseteq k_3) \vee (k_2 \sqsupseteq k_3)).$$

2) Vgl. CARSTENSEN ET AL. (2001), S. 389.

Die Bestimmung der Menge sup(k) aller Superkonzepte zu einem Konzept $k \in K$ hat eine unmittelbare Auswirkung auf die Extension der Superkonzepte. Wie bereits aufgezeigt wurde, müssen die Instanzen eines Konzepts $k \in K$ immer auch in den Extensionen zu allen Konzepten enthalten sein, die zur Menge sup(k) aller Superkonzepte des Konzepts k gehören. Entsprechend müssen die konzeptspezifischen Objektmengen $OB_{Weibliches\_Wesen}$ und OB_{Mensch} aus dem o.a. Beispiel mindestens alle Individuen umfassen, die bereits in OB_{Frau} enthalten sind.

Allerdings scheitert die Präzisierung der intensionalen Semantik sprachlicher Konzepte lediglich auf der Basis der Subkonzeptrelation $\sqsubseteq$ an zwei Punkten.

Erstens wird oftmals im Rahmen der „praktischen Ontologiekonstruktion" darauf verzichtet, alle merkmalsbezogenen Konzepte zu spezifizieren, die zur Beschreibung der intensionalen Semantik von sprachlichen Konzepten notwendig wären. Somit kommt es zu „Definitionslücken", wenn Konzepte nur einigen der Konzepte untergeordnet werden, die in ihrer Gesamtheit hinreichend wären, um dessen intensionale Semantik zu präzisieren. Durch diese „Definitionslücken" werden oftmals nur notwendige, aber nicht hinreichende Merkmale von Konzepten beschrieben.

Zweitens setzt die Präzisierung der intensionalen Semantik von Konzepten durch die Subkonzeptrelation $\sqsubseteq$ voraus, dass die intensionale Semantik der merkmalsbezogenen Konzepte, denen gegenüber das zu beschreibende Konzept taxonomisch untergeordnet ist, bereits feststeht. Die merkmalsbezogenen Superkonzepte eines Konzepts werden allerdings ihrerseits auch durch die taxonomische Unterordnung gegenüber „dritten" Konzepten beschrieben. Die Präzisierung der intensionalen Semantik von Konzepten setzt somit voraus, dass die intensionale Semantik der Superkonzepte bereits eindeutig geklärt ist. Dies muss allerdings nicht immer der Fall sein.

Zur *graphischen Visualisierung* der Subkonzeptrelation bieten sich mehrere zueinander alternative Sprachen an. Hierunter gehören sowohl „traditionelle" Sprachen, wie z.B. Hasse-Diagramme und semantische Netze, als auch aktuell diskutierte Sprachen, wie z.B. die UML. Im Rahmen der werkzeugunterstützten Konstruktion von Ontologien haben sich weitestgehend *hyperbolische* Darstellungen der Subkonzeptrelation durchgesetzt. In der Abbildung 10 ist ein Screenshot der Ontologie-Entwicklungsumgebung *OntoEdit* enthalten, die die Spezifikation der Subkonzeptrelation auf der Basis eines hyperbolischen Graphen unterstützt.

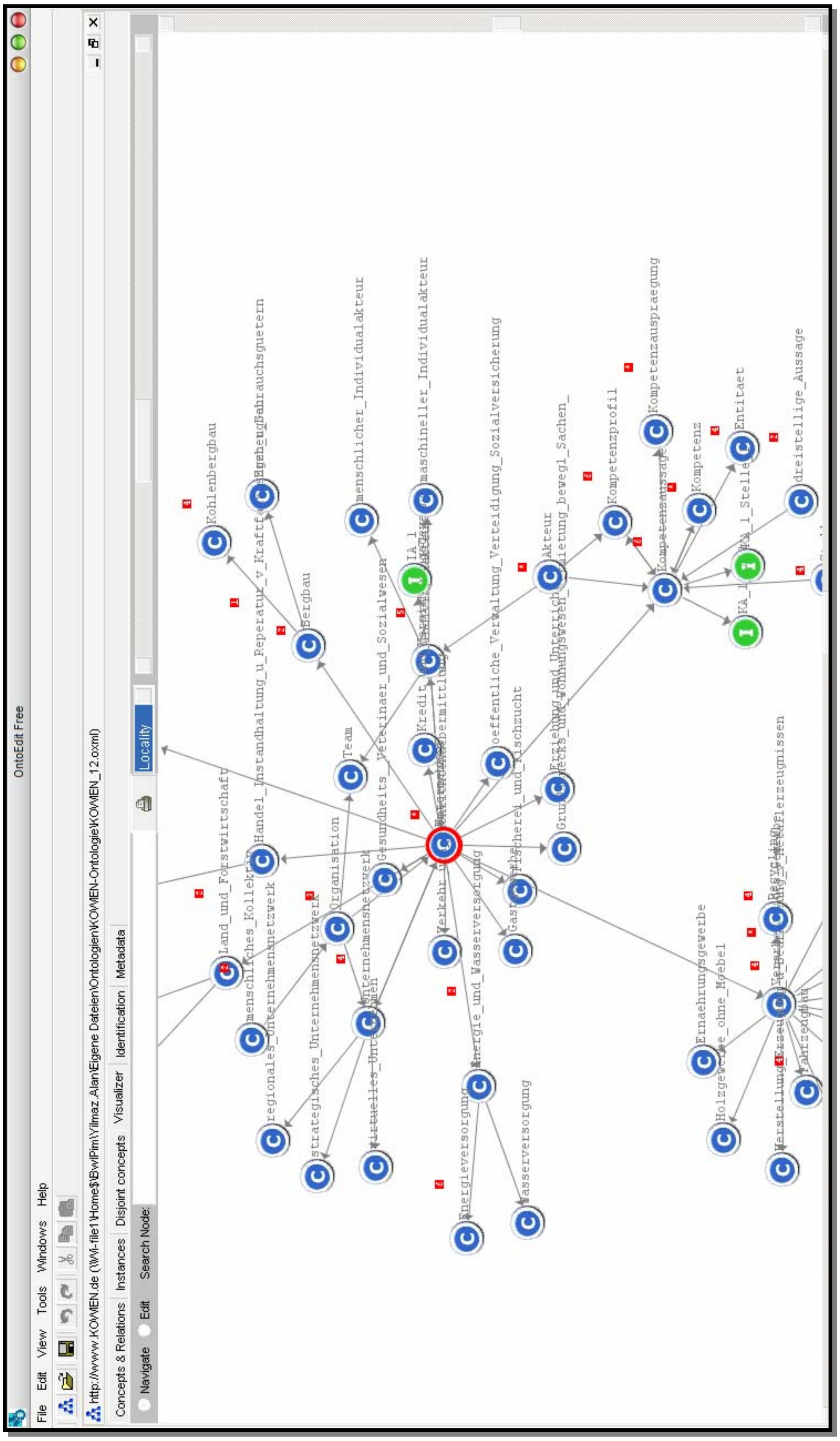

Abbildung 10: Screenshot zur Visualisierung der Subkonzeptrelation in OntoEdit

Die zweite Strukturierungsrelation aus ontologischen Signaturen ist die Inkompatibilitätsrelation $\curlyvee$. Die Inkompatibilitätsrelation $\curlyvee$ ist definiert als:

$$\curlyvee \subseteq (K \times K).$$

Es handelt sich bei $\curlyvee$ um eine *irreflexive* und *symmetrische* Relation:

Irreflexivität $\forall k \in K$: $\neg(k \curlyvee k)$

Symmetrie $\forall k_1,k_2 \in K$: $(k_1 \curlyvee k_2) \leftrightarrow (k_2 \curlyvee k_1)$.

Wenn zwei Konzepte $k_1,k_2 \in K$ in der Inkompatibilitätsrelation $\curlyvee$ zueinander stehen, werden sie als *miteinander inkompatible* Konzepte bezeichnet. Die Menge aller Konzepte, mit denen ein Konzept $k \in K$ inkompatibel ist, wird durch die Funktion

$$\text{ink}: K \rightarrow \text{pot}(K)$$
$$\text{mit ink}(k)=\{k_n \mid (k \curlyvee k_n)\}$$

bestimmt. Über die Inkompatibilität der Konzepte in der Menge ink(k) zu einem Konzept $k \in K$ untereinander wird dadurch keine Aussage getroffen. Die Konzepte, die in ink(k) enthalten sind, können, müssen aber nicht untereinander inkompatibel sein.

Die Extensionen zweier miteinander inkompatibler Konzepte k_1 und k_2 müssen disjunkt sein:

$$\forall k_1,k_2 \in K: (k_1 \curlyvee k_2) \rightarrow OB_{k_1} \cap OB_{k_2} = \varnothing.$$

Demnach dürfen die Konzepte k_1 und k_2 keine gemeinsamen Instanzen aufweisen, wenn sie miteinander inkompatibel sind. Allerdings kann aus dem Nicht-Enthaltensein eines Individuums ob_u in der Extension OB_{k_1} des Konzepts k_1 nicht auf sein Enthaltensein in der Extension OB_{k_2} des Konzepts k_2 geschlossen werden. Bei inkompatiblen Konzepten $k_1,k_2 \in K$ ist nämlich das Nicht-Enthaltensein eines Individuums ob_u in der Extension OB_{k_1} des Konzepts k_1 zwar notwendig, aber nicht hinreichend für das Enthaltensein von ob_u in der Extension OB_{k_2} von k_2. Hingegen ist das Enthaltensein eines Individuums ob_u in der Extension OB_{k_1} des einen Konzepts k_1 hinreichend für das Nicht-Enthaltensein von ob_u in der Extension OB_{k_2} des anderen Konzepts k_2.

Zwei mengenwertige Konzepte $MEN(k_1),MEN(k_2) \in K_{MW}$ sind genau dann miteinander inkompatibel, wenn die einwertigen Konzepte $k_1,k_2 \in K_{EW}$, von denen sie abgeleitet sind, miteinander inkompatibel sind:

$$\forall k_1,k_2 \in K_{EW}, MEN(k_1),MEN(k_2) \in K_{MW}: (k_1 \curlyvee k_2) \rightarrow$$
$$((\exists k_3,k_4 \in K_{MW}: k_3=MEN(k_1) \wedge k_4=MEN(k_2)) \rightarrow k_3 \curlyvee k_4).$$

Wenn nämlich die Extensionen OB_{k_1} und OB_{k_2} zweier einwertiger Konzepte $k_1,k_2 \in K_{EW}$ miteinander disjunkt sein müssen, dann müssen auch die Potenzmengen $pot(OB_{k_1})$ bzw. $pot(OB_{k_2})$ miteinander disjunkt sein.[1)]

1) Wenn

(1.) $OB_{k_1} \cap OB_{k_2}=\varnothing$ wegen $k_1 \curlyvee k_2$, Der Zusammenhang ist in der folgenden Abbildung illustriert. Die Kanten in Abbildung repräsentieren jeweils die Funktionen, denen ihre Annotationen entsprechen. Zwischen den konzeptspezifischen Objektmengen OB_{k_1} und OB_{k_2} bzw. $OB_{MEN(k_1)}$ und $OB_{MEN(k_2)}$ sind jeweils beidseitig gerichtete Kanten eingezeichnet, durch die verdeutlicht wird, dass die Mengen zueinander disjunkt sind.

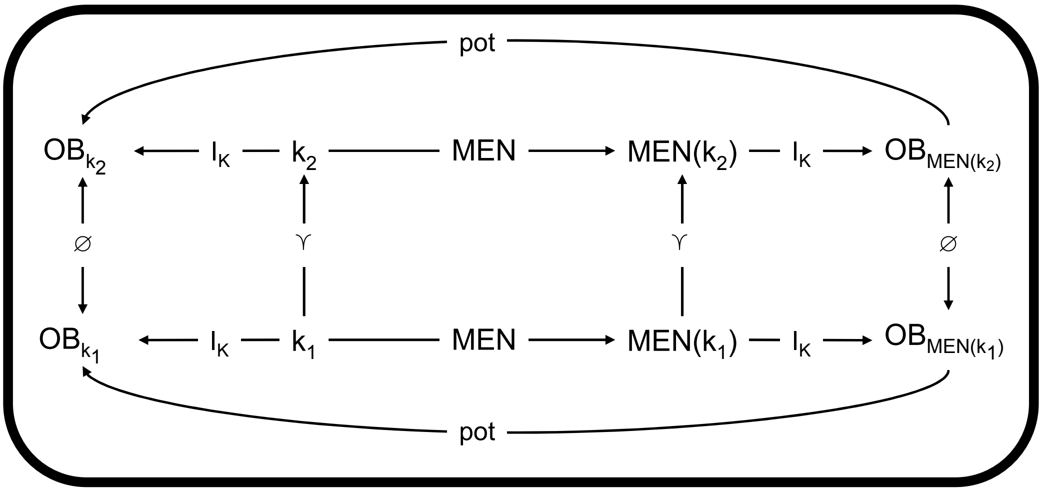

Abbildung 11: Extensionen ein- und mengenwertiger inkompatibler Konzepte

Wenn zwei Konzepte k_1,k_2 in der Inkompatibilitätsrelation $\curlyvee$ zueinander stehen, müssen alle Subkonzepte k von k_1 auch in Inkompatibilitätsrelation zu k_2 stehen :

$$\forall k_1,k_2 \in K: k_1 \curlyvee k_2 \rightarrow (\forall k \in K: (k \sqsubseteq k_1) \rightarrow (k \curlyvee k_2)).$$

Da die Inkompatibilitätsrelation $\curlyvee$ als symmetrische Relation definiert ist, gilt analog, dass alle Subkonzepte k von k_2 in der Inkompatibilitätsrelation mit k_1 stehen müssen. Die Extension OB_{k_1} umfasst nämlich alle formalen Objekte, die in der Extension OB_k zu einem Subkonzept $k \in sub(k_1)$ enthalten sind. Wenn demnach OB_{k_1} disjunkt mit OB_{k_2} ist, muss auch OB_k disjunkt mit OB_{k_2} sein.

(2.) $OB_{MEN(k_1)}=pot(OB_{k_1})=\{TOB \mid TOB \subseteq OB_{k_1}\}$ und

(3.) $OB_{MEN(k_2)}=pot(OB_{k_2})=\{TOB \mid TOB \subseteq OB_{k_2}\}$ gelten, dann gilt für jede Menge

(4.) $TOB \in pot(OB_{k_1})$ aufgrund (1.)

(5.) $TOB \not\subseteq OB_{k_2}$ und somit $TOB \notin pot(OB_{k_2})$.

Die Disjunktheit der Extensionen von inkompatiblen mengenwertigen Konzepten gilt nicht nur dann, wenn die Extensionen jeweils Potenzmengen der Extensionen der zugehörigen einwertigen Konzepte sind. Die Disjunktheit der Extensionen von inkompatiblen mengenwertigen Konzepten muss auch dann gelten, wenn es aufgrund des exponentiellen Wachstums der Mächtigkeit von Extensionen zu mengenwertigen Konzepten ausreicht, dass $OB_{MEN(k_1)}$ bzw. $OB_{MEN(k_2)}$ jeweils *Teilmengen* von $pot(OB_{k_1})$ bzw. $pot(OB_{k_2})$ sind.

Mit $\triangleq$ wird die *Äquivalenzrelation* bezeichnet.[1] Es handelt sich hierbei um eine metasprachliche Strukturierungsrelation, die Beziehungen zwischen Konzepten umfasst, deren Extensionen immer gleich sein müssen. Sie ist als reflexive, symmetrische und transitive Ordnungsrelation definiert:

Reflexivität	$\forall k \in K$:	$(k \triangleq k)$
Symmetrie	$\forall k_1,k_2 \in K$:	$(k_1 \triangleq k_2) \leftrightarrow (k_2 \triangleq k_1)$
Transitivität	$\forall k_1,k_2,k_3 \in K$:	$(k_1 \triangleq k_2) \wedge (k_2 \triangleq k_3) \rightarrow (k_1 \triangleq k_3)$.

Wenn zwei Konzepte $k_1,k_2 \in K$ in Äquivalenzrelation $\triangleq$ zueinander stehen, dann müssen sie mit gleichen[2] konzeptspezifischen Objektmengen extensional interpretiert werden:

$$\forall k_1,k_2 \in K: (k_1 \triangleq k_2) \rightarrow (OB_{k_1} = OB_{k_2}).$$

Die Konzepte k_1 und k_2 werden in diesem Fall auch als (zueinander) *äquivalente* Konzepte bezeichnet.

Der Zusammenhang zwischen äquivalenten Konzepten geht auf die Beobachtung FREGES zurück, dass es Konzepte gibt, deren „Art des Gegebenseins"[3] zwar unterschiedlich ist, die aber extensional stets durch die gleichen Mengen formaler Objekte in-

1) Die konzeptbezogene Äquivalenzrelation $\triangleq$ weist eine Ähnlichkeit zu der *logischen Äquivalenzrelation* $\equiv$ auf, die in Zusammenhang mit prädikatenlogischen und ontologischen Formeln vorgestellt wurde (vgl. Abschnitt 3.1.3.1.2.2). Beide Relationen drücken nämlich die Substituierbarkeit ihrer Relationskomponenten bei gleich bleibender extensionaler Semantik aus. Mit der Äquivalenzrelation $\triangleq$ zwischen Konzepten wird ausgedrückt, dass in allen SIG_{OS}-Strukturen $A_{SIG_{OS}}$, mit denen eine ontologische Signatur SIG_{OS} extensional interpretiert werden kann, die Extensionen OB_{k_1} und OB_{k_2} der äquivalenten Konzepte $k_1,k_2 \in K$ mit $k_1 \triangleq k_2$ gleich sein müssen. Mit der Äquivalenzrelation $\equiv$ zwischen Formeln wird hingegen ausgedrückt, dass der Wahrheitsgehalt einer Formel F_1 in jeder SIG_{OS}-Struktur der gleiche ist, wie der Wahrheitswert einer zu F_1 äquivalenten Formel F_2.

2) Mit der *Gleichheit* zweier konzeptspezifischer Objektmengen ist gemeint, dass jedes Element aus der einen Objektmenge auch in der anderen Objektmenge enthalten sein muss. Dabei kann es sich trotzdem um zwei nicht-*identische* Objektmengen handeln. Als Unterscheidungskriterium dienen die Namen der konzeptspezifischen Objektmengen. Der einzige Unterfall der Gleichheit ist die *Identität*; vgl. WEDEKIND ET AL. (2004), S. 337. Im Fall identischer konzeptspezifischer Objektmengen sind auch die Namen der Objektmengen gleich.

3) FREGE (1969), S. 45.

terpretiert werden.[1] Durch die Spezifikation der Äquivalenz von Konzepten wird darüber Wissen ausgedrückt, dass zwei Konzepte aufgrund domänenspezifischer Umstände zusammenhängen. Zwar entspricht jedes (sprachliche) Konzept aus einer Ontologie einer einzigartigen Intension, da es eineindeutig einem mentalen Konzept zugeordnet werden kann. Allerdings sind die Intensionen der beiden Konzepte derart miteinander „verdrahtet“, dass die Extensionen gleich sein müssen. Beispiele für die Äquivalenz von Konzepten wurden eingangs im Rahmen des Anforderungskatalogs für die statische Struktur des Modellierungskonzepts aufgezeigt.[2]

Die Bestimmung aller Konzepte, die mit einem Konzept $k \in K$ äquivalent sind, erfolgt durch die Funktion

$$\text{aeq}: K \rightarrow \text{pot}(K)$$
$$\text{mit } \text{aeq}(k)=\{k_n \mid (k_n \circeq k)\}.$$

Da es sich bei der Äquivalenzrelation $\circeq$ um eine symmetrische und zugleich transitive Relation handelt, müssen die Mengen $\text{aeq}(k_1)$ und $\text{aeq}(k_2)$ gleich sein, wenn $(k_1 \circeq k_2)$ gilt. Analog sind die Mengen $\text{aeq}(k_1)$ und $\text{aeq}(k_2)$ disjunkt, wenn $(k_1 \circeq k_2)$ nicht gilt.

---

1) Als anschauliches Beispiel diente FREGE die altertümliche Vermutung, bei dem hellsten Stern am Morgenhimmel würde es sich um einen anderen Planeten handeln als um den hellsten Stern am Abendhimmel. Während dem erstgenannten mentalen Konzept das sprachliche Konzept *Morgenstern* zugeordnet wurde, wurde dem zweitgenannten mentalen Konzept das sprachliche Konzept *Abendstern* zugeordnet. Mit der Feststellung, dass den beiden unterschiedlichen mentalen Konzepten als gemeinsame Extension der heute als „Venus“ bekannte Planet zugewiesen werden kann, musste die Vermutung revidiert werden. Ein weiteres Beispiel äquivalenter Konzepte gibt QUINE mit den sprachlichen Konzepten *Everest* und *Gausiankar*; vgl. QUINE (1969), S. 257. Die Äquivalenz der beiden sprachlichen Konzepte ist damit begründet, dass ihnen zwar unterschiedliche mentale Konzepte als Intensionen zugeordnet wurden, deren Extensionen allerdings beide den Berg „Mount Everest“ aus unterschiedlichen geographischen Beobachtungsrichtungen umfassen. Es handelt sich hierbei wohlgemerkt um keine *Synonymie* in dem hier vorgelegten Verständnis. Zwar unterscheiden sich sowohl FREGEs als auch QUINEs Synonymie-Konzeptionen von der vorliegenden Arbeit. Bei beiden Autoren wird nämlich die Synonymie auf der Ebene der sprachlichen Konzepte definiert. In der vorliegenden Arbeit erfolgt hingegen eine Differenzierung zwischen (objekt-)sprachlichen Konzepten und metasprachlichen Bezeichnern. Synonymie-Beziehungen können in der vorliegenden Arbeit nur zwischen metasprachlichen Bezeichnern existieren. Auf diesen Aspekt wird in den folgenden Abschnitten näher eingegangen. In den oben wiedergegebenen Beispielen unterscheiden sich allerdings auch die mentalen Konzepte, die den sprachlichen Konzepten zugeordnet werden, voneinander. Am Ehesten lassen sich „Identitätssätze“ der Form „Everest=Gaurisankar“ und „Abendstern=Morgenstern“ (vgl. QUINE (1969), S. 268) mit dem hier vorgelegten Ansatz der Äquivalenz von Konzepten vereinbaren. Damit werden nämlich die Identitäten der Objekte ausgedrückt, die in den Extensionen der jeweiligen sprachlichen Konzepte enthalten sind. Zwar sind in QUINEs Beispielen lediglich solche sprachlichen Konzepte aufgeführt, deren Extensionen immer nur ein Objekt umfassen. Allerdings lässt sich das Prinzip äquivalenter Konzepte auch auf solche sprachlichen Konzepte ausweiten, deren Extensionen mehrere Objekte umfassen. Ein Beispiel hierfür wurde eingangs mit den Konzepten *Aufgaben_Abteilung_A* und *Aufgaben_ Mitarbeiter_4711* vorgestellt.

2) Vgl. Abschnitt 2.1.3.2.2.

3.1.3.2.2.2 Metasprachliches Alphabet

Für die Zuordnung einer extensionalen Semantik wird in der vorliegenden Arbeit der Ansatz der sortierten Prädikatenlogik beibehalten. Die extensionale Semantik von objektsprachlichen Komponenten aus einer ontologischen Signatur SIG_{OS} wird demnach durch jene Konstrukte aus einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ gegeben, auf die die jeweiligen Komponenten mittels einer Interpretationsfunktion aus IF_{OS} abgebildet werden. Somit ist die extensionale Semantik ontologischer Signaturen durchgehend *formal* bestimmt.

Darüber hinaus wird mit der Konstruktion von Ontologien auch das Ziel verfolgt, sprachlichen Konstrukten eine zumindest für Menschen verarbeitbare, *intensionale* Semantik zuzuordnen. Die Zuordnung einer intensionalen Semantik zu den objektsprachlichen Komponenten einer ontologischen Signatur SIG_{OS} erfolgt u.a.[1)] durch die Mitglieder der Familie bezf sprachspezifischer Bezeichnungsfunktionen und die Mitglieder der Familie deff sprachspezifischer Definitionsfunktionen. Im Gegensatz zu der extensionalen Semantik, die durchgehend formal festgelegt ist, ist allerdings die intensionale Semantik durch Bezeichnungs- und Definitionsfunktionen *informal* bestimmt. Dabei ist es für beide Funktionen notwendig, auf ein *natürlichsprachliches Alphabet* zurückzugreifen, um mit dessen Elementen sprachspezifische Bezeichner bzw. Definitionen konstruieren zu können.

Für diese Zwecke wird das meta- und natürlichsprachliche Alphabet $ALPH_{META}$ herangezogen. Es wird zum einen dazu verwendet, den deskriptiven Symbolen aus einer ontologischen Signatur SIG_{OS} natürlichsprachliche *Bezeichnungen* zuzuordnen. Bei den Bezeichnungen der deskriptiven Symbole werden ihnen natürlichsprachliche Zeichenketten über $ALPH_{META}$ zugeordnet. Dabei erfolgt jede Bezeichnung stets in Bezug auf eine natürliche Sprache. Zum anderen wird das natürlichsprachliche Alphabet $ALPH_{META}$ bei der Zuordnung sprachspezifischer *Definitionen* zu den deskriptiven Symbolen verwendet. Auch die Definitionen sind natürlichsprachliche Zeichenketten, die stets in Bezug auf eine natürliche Sprache bestimmt sind. Darüber hinaus werden beim Übergang von ontologischen Signaturen zu ontologischen Spezifikationen Regeln eingeführt, denen ebenso jeweils sprachspezifische Definitionen zugeordnet werden können.

Die Festlegung des natürlichsprachlichen Alphabets $ALPH_{META}$ und der Menge $ALPH_{META}*$ aller Zeichenketten über $ALPH_{META}$ orientiert sich an dem Aufbau formaler Sprachen.[2)] Das metasprachliche Alphabet $ALPH_{META}$ umfasst die metasprachlichen *Zeichen*:

---

1) Über die Zuordnung von natürlichsprachlichen Bezeichnern und Definitionen zu objektsprachlichen Komponenten hinaus wird ihre intensionale Semantik insbesondere durch Strukturierungsrelationen bestimmt. Auf diesen Aspekt der intensionalen Semantik objektsprachlicher Komponenten ontologischer Signaturen wurde bereits eingegangen.

2) Vgl. dazu CARSTENSEN ET AL. (2001), S. 59 f.; EHRIG ET AL. (1999), S. 23 ff.; ERK/PRIESE (2002), S. 27 ff.

$$ALPH_{META}=\{\texttt{0,1,...,9,a,b,...,z,A,B,...,Z,,,;,.,:,+,-,?,!,\$,\%;/,(,),=,\ }\}.$$

Jede Folge $z_1...z_n$ mit $z_1,..,z_n \in ALPH_{META}$ ist eine *Zeichenkette* der Länge n über dem Alphabet. Insbesondere gilt, dass das ausgezeichnete Element λ eine Zeichenkette der Länge n=0 über dem Alphabet ist, obwohl es nicht als Zeichen in $ALPH_{META}$ enthalten ist. Es darf nicht mit dem leeren Zeichen „ “ verwechselt werden, das auch im Alphabet $ALPH_{META}$ enthalten ist.[1)]

Die Menge $ALPH_{META}*$ aller metasprachlichen Zeichenketten über $ALPH_{META}$ ist induktiv definiert durch:

(1.) $\lambda \in ALPH_{META}*$

(2.) $\forall z \in ALPH_{META}: z \in ALPH_{META}*$

(3.) $\forall z_1 \in ALPH_{META}, z_2 \in ALPH_{META}*: z_1 z_2 \in ALPH_{META}*$

Die Induktionsbasen der oben angegeben Definition der Menge $ALPH_{META}*$ aller Wörter über dem Alphabet $ALPH_{META}$ sind zum einen die leere Zeichenkette λ und zum anderen alle Zeichen aus dem Alphabet. Sowohl λ als auch jedes Zeichen aus der Menge $ALPH_{META}$ stellen nämlich atomare Zeichenketten[2)] aus der Menge $ALPH_{META}*$ dar.

Die Menge $ALPH_{META}*$ aller Zeichenketten über dem Alphabet $ALPH_{META}$ kann folgendermaßen in paarweise zueinander disjunkte Teilmengen unterteilt werden:

$$ALPH_{META}* = ALPH_{META}^0 \cup ALPH_{META}^1 \cup ... \cup ALPH_{META}^N.$$

Jede Teilmenge $ALPH_{META}^n$ umfasst Zeichenketten der Länge n und wird folgendermaßen definiert:[3)]

(1.) $ALPH_{META}^0=\{\lambda\}$,

(2.) $ALPH_{META}^1=\{z \mid z \in ALPH_{META}\}$,

(3.) $ALPH_{META}^n=\{z_1 z_2 \mid z_1 \in ALPH_{META} \wedge z_2 \in ALPH_{META}^{n-1}\}$
für $n \geq 2$ und $n \in \mathbb{N}_+$.

Die Länge n einer Zeichenkette $z \in ALPH_{META}*$ lässt sich als Wert der Funktion

1) In der Praxis wird statt des leeren Zeichens oftmals das Zeichen „_“ verwendet. Auch in der vorliegenden Arbeit werden in Bezeichnern, die sich aus mehreren Wörtern zusammensetzen, die Wörter durch „_“ verbunden. Aus Zwecken der Kompatibilität mit dem natürlichen Sprachgebrauch wird allerdings für die Zeichenketten über $ALPH_{META}$ hiervon abgesehen.

2) Von *atomaren* Zeichen*ketten* zu sprechen wirkt zwar kontraintuitiv, da mit Ketten in der Regel auch eine Zerlegbarkeit in ihre Glieder (i.d.F. Zeichen) assoziiert wird. Daher werden Zeichenketten oftmals in der Literatur als *Wörter* bezeichnet; vgl. die zuvor angegebenen Quellen zur Konstruktion formaler Sprachen. Allerdings wird vom Verfasser die Bezeichnung *Zeichenkette* gegenüber der Bezeichnung *Wort* präferiert, da die Bilder der sprachspezifischen Definitionsfunktion def_{lan} *Sätze* im intuitiven Verständnis sind. Dies wird in späteren Abschnitten vorgesellt. Die Bezeichnung *Zeichenkette* ist zur Erfassung von Definitionen besser geeignet als die Bezeichnung *Wort*.

3) Vgl. EHRIG ET AL. (1999), S. 24; ZELEWSKI (1995), Bd. 4, S. 6.

$$len_{ALPH}: ALPH_{META}^* \rightarrow \mathbb{N}$$

bestimmen, wobei

(1.) $len_{ALPH}(\lambda)=0$,

(2.) $len_{ALPH}(z)=1$ wenn $z \in ALPH_{META}$ und

(3.) $len_{Alph}(z_1 z_2)=len_{ALPH}(z_2)+1$, wenn $z_1 \in ALPH_{META}$ gilt.

Für jede Menge $ALPH_{META}^n$ gilt demnach $ALPH_{META}^n=\{z \mid len_{ALPH}(z)=n\}$.

Die Menge $ALPH_{META}^*$ aller Zeichenketten über dem Alphabet $ALPH_{META}$ ist von der Potenzmenge pot($ALPH_{META}$) der Menge $ALPH_{META}$ zu unterscheiden. Während die Menge $ALPH_{META}^*$ solche Zeichenketten beinhaltet, auf deren atomare Komponenten nicht unmittelbar zugegriffen werden kann, umfasst die Potenzmenge pot($ALPH_{META}$) alle Teilmengen der Menge $ALPH_{META}$. Die Elemente der Menge pot($ALPH_{META}$) sind selbst Mengen, deren Elemente wiederum Zeichen aus dem Alphabet $ALPH_{META}$ sind. Die Elemente der Potenzmenge pot($ALPH_{META}$) umfassen Zeichen aus $ALPH_{META}$ auf ungeordnete Weise. In den Elementen der Menge $ALPH_{META}^*$ sind die Zeichen hingegen geordnet.

Im weiteren Verlauf wird auch die Potenzmenge pot($ALPH_{META}^*$) der Menge $ALPH_{META}^*$ aller metasprachlichen Zeichenketten benötigt. Es handelt sich dabei um jene Menge, die alle Teilmengen der Menge $ALPH_{META}^*$ umfasst. Jedes Element der Menge pot($ALPH_{META}^*$) ist somit eine Menge von Zeichenketten, die über dem Alphabet $ALPH_{META}$ konstruiert werden können, oder die leere Menge $\varnothing$.

Abschließend wird darauf aufmerksam gemacht, dass grundsätzlich weder das Alphabet $ALPH_{META}$ noch die daraus abgeleiteten Zeichenketten für ontologische Signaturen notwendig sind. Für die Zwecke, bei denen im Folgenden insbesondere die Menge $ALPH_{META}^*$ verwendet wird, ließe sich nämlich auch die Menge OB_{String} verwenden, mit der das Datenkonzept $String \in K_{DT}$ extensional interpretiert wird. Die Individuen aus der Objektmenge OB_{String} sind nämlich auch stets Zeichenketten. Somit ließen sie sich für die Zwecke, zu denen Zeichenketten über dem Alphabet $ALPH_{META}$ benötigt werden, ebenso verwenden.

Allerdings hat die oben vorgestellte Vorgehensweise einige Vorteile gegenüber der Erfassung von Zeichenketten aus der konzeptspezifischen Objektmenge OB_{String}. Mit der oben vorgelegten Vorgehensweise wird z.B. die Möglichkeit offen gehalten, *sprachspezifische Alphabete* zu definieren.[1] Jedes sprachspezifische Alphabet $ALPH_{lan}$ mit lan={ger,eng,fr,...} würde in diesem Fall sprachspezifische Zeichen beinhalten. Beispielsweise würden im sprachspezifischen Alphabet $ALPH_{fr}$ über die Zeichen hinaus, die im sprachspezifischen Alphabet $ALPH_{ger}$ enthalten sind, u.a. auch noch die Zeichen é, è und ê enthalten sein. Um eine solche Vorgehensweise analog erfassen zu können,

1) Zu sprachspezifischen Alphabeten im Rahmen der konzeptionellen Modellierung vgl. BOMAN ET AL. (1997), S. 21 f.

müsste das Datenkonzept $\mathrm{Char} \in K_{DT}$ weiter in sprachspezifische Datenkonzepte unterteilt[1)] werden. Jedes sprachspezifische Datenkonzept $Char_{ger}$, $Char_{eng}$ usw. würde mit einer solchen konzeptspezifischen Objektmenge interpretiert werden, die nur Zeichen beinhaltet, die in der jeweiligen natürlichen Sprache gültig sind. Entsprechen der Unterteilung des Datenkonzepts *Char* müsste auch das Datenkonzept *String* unterteilt werden. Demnach müssten z.B. $OB_{String_{ger}}=OB_{Char_{ger}}*$ und $OB_{String_{eng}}=OB_{Char_{eng}}*$ gelten. Eine solche Unterteilung ist allerdings weder für das Datenkonzept *Char* noch für das Datenkonzept *String* üblich. Die Unkonventionalität einer Unterteilung des Datenkonzepts String ist jedoch kein Argument, das den Ausschluss dieser Variante erfordern würde. Schließlich wurde in der vorliegenden Arbeit bereits eine Unterteilung von Datenkonzepten vorgestellt, um die hierarchische Beziehung zwischen Datenkonzepten ausdrücken zu können.[2)]

Vielmehr ist mit der oben vorgestellten Vorgehensweise der Versuch verbunden, eine intensionale Semantik für die Komponenten ontologischer Signaturen anzugeben, die nicht auf eine extensionale Semantik – wie es bei den Zuordnungen der Objektmengen OB_{Char} und OB_{String} zu den Datenkonzepten Char bzw. String der Fall wäre – zurückgreift. Die Zeichenketten werden nämlich im weiteren Verlauf dazu verwendet, Bezeichnungen und Definitionen für die objektsprachlichen Komponenten ontologischer Signaturen anzugeben. Darüber hinaus werden bestimmten ontologischen Formeln natürlichsprachliche Erläuterungen zugeordnet. Würden nun für diese Zwecke Zeichenketten aus konzeptspezifischen Objektmengen verwendet, würde eine – zumindest partielle – extensionale Interpretation von Konzepten vorausgesetzt werden. Es müsste nämlich mindestens das Datenkonzept $\mathrm{String} \in K_{DT}$ extensional durch die konzeptspezifische Objektmenge OB_{String} interpretiert werden, um die formalen Objekte aus der Menge OB_{String} für die oben angegebenen Zwecke verwenden zu können.

Die Argumentation gegen eine Verwendung der Elemente von OB_{String} ist analog zu der Argumentation gegen eine Verwendung der Objektmenge OB_{Bool} zur Erfassung von Wahrheitswerten. Als Wahrheitswerte von Formeln werden nicht die Elemente der konzeptspezifischen Objektmenge OB_{Bool} zugeordnet, da der Begriff der *Wahrheit* eine besondere Bedeutung in logischen Analysen hat. Würden Objekte aus der Menge OB_{Bool} zur Kennzeichnung der Gültigkeit von Formeln über ontologischen Signaturen herangezogen werden, wäre auch hier eine – zumindest partielle – extensionale Interpretation der ontologischen Signatur SIG_{OS} notwendig. Um die Wahrheitswerte von Formeln auch ohne die extensionale Interpretation des Datenkonzepts *Bool* bestimmen zu können, wurden die Formeln w und f als formale Konstrukte sui generis eingeführt. Aus dem gleichen Grund werden für die Zwecke der Bezeichnung und Definition von objektsprachlichen Konstrukten das Alphabet $ALPH_{META}$ und die Menge $ALPH_{META}*$ der Zeichenketten, die über diesem Alphabet definiert sind, verwendet

1) Die Unterteilung würde mittels der Subkonzeptrelation $\sqsubseteq$ erfolgen.

2) Vgl. Abschnitt 3.1.3.2.2.1

3.1.3.2.2.3 Metasprachliche Bezeichnungs- und Definitionsfunktionen

Es wurde bereits in Abschnitt 3.1.3.2.1.1 darauf hingewiesen, dass für Akteure, die an der Konstruktion einer Ontologie beteiligt gewesen sind, ein bijektiver Zusammenhang zwischen sprachlichen Konzepten aus der Ontologie einerseits und mentalen Konzepten andererseits existiert.[1] Die sprachlichen Konzepte dienen somit lediglich der eindeutigen Identifikation von mentalen Konzepten.

Für Akteure, die an dem Einigungsprozess bei der Konstruktion einer Ontologie nicht beteiligt waren, kann es sich allerdings bei den sprachliche Konzepten um Konstrukte handeln, die sie nicht verstehen. Das heißt, dass möglicherweise Akteure den sprachlichen Konzepten keine eigenen mentalen Konzepte zuordnen können oder sogar solche mentalen Konzepte zuordnen, die von den Akteuren, die an der Ontologiekonstruktion beteiligt waren, nicht intendiert gewesen sind. In diesen Fällen könnten Benutzer der Ontologie den deskriptiven Symbolen entweder keine oder eine „falsche" intensionale Semantik zuordnen.

Um fehlende oder falsche Sinnzuweisungen zu objektsprachlichen Konstrukten möglichst zu vermeiden, verfügen Ontologien über metasprachliche Ausdrucksmittel. Es handelt sich hierbei um die Mitglieder der Familie bezf *sprachspezifischer Bezeichnungsfunktionen* und die Mitglieder der Familie deff *sprachspezifischer Definitionsfunktionen*. Mit Hilfe der genannten Funktionen werden durchgehend formalsprachliche Ausdrucksmittel aus einer ontologischen Signatur SIG_{OS} mit natürlichsprachlichen Ausdrücken verbunden. Eine solche Verbindung trägt einerseits dazu bei, mit Hilfe von Ontologien den Anschluss an die natürliche Begriffswelt zu wahren. Während nämlich die sprachlichen Konzepte stets in eineindeutiger Beziehung zu mentalen Konzepten stehen, können im natürlichen Sprachgebrauch oftmals verschiedene Formen der *Mehrdeutigkeit* oder *Ambiguität* beobachtet werden.[2]

Von den Arten potenzieller Mehrdeutigkeiten sind insbesondere die *Synonymie* und die *Homonymie* von besonderer Bedeutung. Im Fall der Synonymie werden für das gleiche Konzept unterschiedliche Zeichenketten als Bezeichner des Konzepts verwendet. Im Fall der Homonymie fungiert eine Zeichenkette als Bezeichner unterschiedlicher Konzepte. Somit sind sowohl Synonymie als auch Homonymie als Eigenarten der Zeichenketten definiert, mit denen Konzepte bezeichnet werden. Da es sich bei den Bezeichnern

1) Dieser Zusammenhang geht in der Diktion dann unter, wenn beispielsweise von einem Konzept „k" gesprochen wird. Bei einer präziseren Diktion müsste gesagt werden: „das (mentale) Konzept, das durch das sprachliche Konzept „k" bezeichnet wird". Dadurch würde allerdings die Referenz des Begriffs *Konzept* auf eine außersprachliche Ebene verlagert werden, auf die im Rahmen der formalen Semantik nicht zugegriffen werden kann. Außersprachliche Entitäten finden nämlich in der formalen Semantik des hier vorgelegten Ansatzes keine Berücksichtigung. Sie müssen – in einem eventuell nachgelagerten Schritt – durch eine denotationale Semantik beschlossen werden.

2) Vgl. LÖBNER (2003), S. 53 ff.

der sprachlichen Konzepte um metasprachliche Konstrukte handelt, werden sowohl Synonymie als auch Homonymie auf metasprachlicher Ebene festgelegt.

Weder die Homonymie noch die Synonymie werden somit in ontologischen Signaturen auf der *objektsprachlichen* Ebene berücksichtigt. Sowohl Homonymie als auch Synonymie werden als Beziehungen zwischen *metasprachlichen* Bezeichnern spezifiziert. Würde die Homonymie als Eigenart sprachlicher Konzepte zugelassen werden, müsste die denotationale Interpretation von sprachlichen Konzepten durch mentale Konzepte rechtsmehrdeutig sein.[1)] Würde die Synonymie als Beziehung zwischen sprachlichen Konzepten definiert werden, wäre die denotationale Interpretation linksmehrdeutig. Es wurde allerdings bereits darauf hingewiesen, dass die Eineindeutigkeit der denotationalen Interpretationsfunktion ein wesentliches Charakteristikum ontologiegestützter Modellierung ist. Daher werden sowohl homonyme als auch synonyme sprachliche Konzepte ausgeschlossen.[2)]

Trotz der Festlegung auf die eineindeutige denotationale Interpretation von sprachlichen Konzepten erweisen sich Ontologien als äußerst flexibel in Bezug auf die Berücksichtigung von Mehrdeutigkeiten.[3)] Aufgrund dieser Flexibilität heben sich Ontologien deutlich von traditionellen und aktuell diskutierten Modellierungssprachen ab. In der Regel werden nämlich von Modellierungssprachen keine Ausdrucksmittel zur Verfügung gestellt, um die in der natürlichen Sprache vorkommenden Mehrdeutigkeiten auch in den entsprechenden Modellen berücksichtigen zu können.

In Ontologien können hingegen Mehrdeutigkeiten dadurch berücksichtigt werden, dass einerseits formal- und objektsprachliche Konstrukte[4)] und andererseits natürlich- und metasprachliche Konstrukte voneinander unterschieden werden. Beispielsweise werden sprachliche Konzepte weiterhin als elementare formale Bausteine von Ontologien beibehalten. Demgegenüber können allerdings in Ontologien sprachlichen Konzepten auch natürlichsprachliche Bezeichner zugeordnet werden. Die Bezeichner von sprachlichen Konzepten werden dabei für die Rekonstruktion der natürlichsprachlichen Begriffswelt

---

1) Die Rechtsmehrdeutigkeit gilt genau dann, wenn die Interpretation der sprachlichen Konzepte durch mentale Konzepte betrachtet wird. Wenn hingegen die Referenzbeziehung von mentalen zu sprachlichen Konzepten betrachtet wird, handelt es sich um eine linksmehrdeutige Relation. Für die folgenden Ausführungen wird allerdings stets von der ersten Relation ausgegangen.

2) Zur hierzu alternativen Zulässigkeit synonymer deskriptiver Symbole vgl. PATIG (2004), S. 99. Um die Extensionsgleichheit von Konzepten ausdrücken zu können, wurde in der vorliegenden Arbeit die Äquivalenzrelation $\triangleq$ eingeführt. Äquivalenzen von Operations- und Relationssymbolen lassen sich hingegen am einfachsten mit Hilfe *objektsprachlicher Inferenzregeln* ausdrücken, auf die in Abschnitt 3.1.4.1 eingegangen wird.

3) Zu den Möglichkeiten, in Ontologien natürlichsprachliche Mehrdeutigkeiten zu berücksichtigen, vgl. PREECE ET AL. (2001), S. 183 ff.

4) Zu beachten ist, dass die Berücksichtigung von Mehrdeutigkeiten sich in Ontologien nicht nur auf Konzepte erstreckt. Die Berücksichtigung erstreckt sich ebenso auf Operations- und Relationssymbole und somit auf alle deskriptiven Symbole.

verwendet.[1)] Dadurch wird eine Separation zwischen dem formalsprachlichen Apparat einer Ontologie einerseits und der natürlichsprachlichen Begriffswelt andererseits bewirkt. Während der erste Aspekt der Formalsprachlichkeit sowohl notwendig als auch hinreichend für die computergestützte Analyse von ontologiegestützten Modellen ist, wird mit dem zweiten Aspekt – der Berücksichtigung natürlichsprachlicher Begriffswelten – gewährleistet, dass menschliche Ontologienutzer am „Front-End“ von Ontologien sich nicht mit formalsprachlichen Konstrukten auseinander setzen müssen.

Die Familie bezf der sprachspezifischen Bezeichnungsfunktionen

$$\text{bezf}=(\text{bez}_{\text{ger}}, \text{bez}_{\text{eng}}, \text{bez}_{\text{fr}},...)$$

hat als Mitglieder Funktionen, mit denen deskriptiven Symbolen aus einer ontologischen Signatur SIG_{OS} jeweils eine nicht-leere Menge natürlichsprachlicher *Bezeichner* zugeordnet werden können.[2)] Es handelt sich dabei um Zeichenketten über dem natürlichsprachlichen Alphabet $ALPH_{META}$. Der Index lan∈{ger,eng,fr,...} gibt an, welcher natürlichen Sprache die Bezeichner entstammen. Exemplarisch wurden hier Deutsch (ger), Englisch (eng) und Französisch (fr) aufgezählt. Die Mengenfamilie lässt sich problemlos ausweiten, um auch andere natürliche Sprachen berücksichtigen zu können. Wenn die jeweilige natürliche Sprache im Argumentationskontext nicht von Relevanz ist, wird im Folgenden auch von einer Bezeichnungsfunktion bez_{lan} gesprochen.

Jede sprachspezifische Bezeichnungsfunktion bez_{lan} ist definiert als

$$\text{bez}_{\text{lan}}: (K \cup OPS \cup RS) \rightarrow \text{pot}_{+}(ALPH_{META}{}^{*}).$$

Die Bildmenge der Bezeichnungsfunktion ist eine echte und nicht-leere Teilmenge der Menge $ALPH_{META}{}^{*}$. Die Menge $ALPH_{META}{}^{*}$ wurde bereits zuvor als die Menge aller Zeichenketten über dem Alphabet $ALPH_{META}$ vorgestellt.

Bei den sprachspezifischen Bezeichnungsfunktionen bez_{lan} handelt es sich um totale Funktionen. Es müssen nämlich allen deskriptiven Symbolen aus einer ontologischen Signatur SIG_{OS} natürlichsprachliche Bezeichnungen zugewiesen werden. In der Regel enthält die Menge $\text{bez}_{\text{lan}}(x)$ zu einem deskriptiven Symbol $x \in (K \cup OPS \cup RS)$ mindestens x in seiner „natürlichsprachlichen Variante“ als $\texttt{x} \in \text{bez}_{\text{lan}}(x)$. Beispielsweise kann die Bezeichnermenge zu einem Konzept *Unternehmen* lauten:

$$\text{bez}_{\text{ger}}(\text{Unternehmen})=\{\texttt{Unternehmen},\texttt{Unternehmung},\texttt{Betrieb}\}.$$

---

1) Dies schließt nicht aus, dass auch die objektsprachliche Ebene einer Ontologie für die Rekonstruktion der natürlichsprachlichen Begriffswelt verwendet werden kann. Es bietet sich sogar an, solche deskriptiven Symbole aus einer ontologischen Signatur SIG_{OS} zu verwenden, die natürlichsprachlichen Begriffen entsprechen. Allerdings können auf dieser Ebene keine Mehrdeutigkeiten berücksichtigt werden.

2) Vgl. MÄDCHE (2002), S. 18. („lexical entries“)

Darüber hinaus ist jede sprachspezifische Bezeichnungsfunktion bez_{lan} nicht notwendig *injektiv*. Das heißt, dass zwei unterschiedlichen Konzepten $k_1,k_2 \in K$ die gleiche Menge von Zeichenketten als Bezeichner zugeordnet werden kann.[1)]

Mit sprachspezifischen Bezeichnungsfunktionen ist es möglich, Gemeinsamkeiten und Unterschiede zwischen den Bezeichnern von Konzepten formal zu präzisieren. In erster Linie handelt es sich hierbei um die bereits erwähnten Phänomene der *Homonymie* und *Synonymie*. Homonymie und Synonymie werden gemeinsam als *Bezeichnerbeziehungen* angesprochen. Welche Bezeichnerbeziehung bei welcher Konstellation der Bezeichnungs- und Definitionsfunktion vorliegt, kann der Tabelle 9 entnommen werden:

	Bezeichnerbeziehung	
	Homonymie	Synonymie
deskriptive(s) Symbol(e)	$x_1 \neq x_2$	x
Bezeichner	$b \in bez_{lan_1}(x_1) \wedge b \in bez_{lan_2}(x_2)$	$b_1 \in bez_{lan_1}(x) \wedge b_2 \in bez_{lan_2}(x)$

Tabelle 9: Homonymie und Synonymie

Ein *Homonym* ist eine Zeichenkette, die für unterschiedliche deskriptive Symbole gilt, die stets unterschiedliche Intensionen und möglicherweise unterschiedliche Extensionen haben.[2)] Es wird in der zweiten Spalte der Tabelle 9 vorgestellt. Im Fall von homonymen Konzeptbezeichnern werden zwei unterschiedliche Konzepte $k_1,k_2 \in K$ jeweils u.a. mit einer Zeichenkette b bezeichnet. Es können z.B. sowohl das Konzept *Parkbank* als auch das Konzept *Geldinstitut* mit der Zeichenkette `Bank` bezeichnet sein. Da es sich um zwei unterschiedliche Konzepte handelt, müssen sie auch jeweils eine unterschiedliche Intension aufweisen. In der Regel kann sogar angenommen werden, dass die Extensionen $OB_{Parkbank}$ und $OB_{Geldinstitut}$ disjunkt sind.

1) Auf den folgenden Seiten werden Sonderfälle behandelt, die sich aus der Gleichheit der Bezeichner für unterschiedliche Konzepte (Homonymie) ergeben können.

2) Vgl. LÖBNER (2003), S. 58 ff.; LUTZEIER (1995), S. 33; ORTNER (1997), S. 32.

Eine Zeichenkette $b \in ALPH_{META}*$ ist Element der Menge

$$HOM \subseteq ALPH_{META}^{*},$$

wenn b ein Homonym ist.[1)] b ist wiederum genau dann ein Homonym, wenn es zwei unterschiedlichen deskriptiven Symbole $x_1,x_2 \in (K \cup OPS \cup RS)$ mit $x_1 \neq x_2$ als natürlichsprachlicher Bezeichner zugeordnet ist. Es gilt demnach für jede natürliche Sprache $lan_1,lan_2 \in \{ger,eng,fr,...\}$:

$$\forall b \in ALPH_{META}*: HOM(b) \leftrightarrow$$
$$\exists x_1,x_2 \in (K \cup OPS \cup RS): x_1 \neq x_2 \wedge b \in bez_{lan_1}(x_1) \wedge b \in bez_{lan_2}(x_2).$$

Dabei ist zu beachten, dass die sprachspezifischen Bezeichnungsfunktionen bez_{lan_1} und bez_{lan_2} nicht notwendig unterschiedlich sein müssen. Es lassen sich hierbei zwei Fälle unterscheiden.

Wenn es sich im ersten Fall bei den beiden Funktionen aufgrund $lan_1=lan_2$ um *eine* sprachspezifische Bezeichnungsfunktion $bez_{lan_1}=bez_{lan_2}$ handelt, liegt eine *Homonymie i.e.S.* vor. Demnach wird bei der Homonymie i.e.S. zwei unterschiedlichen deskriptiven Symbolen *ein* Bezeichner in *einer* natürlichen Sprache zugeordnet, wenn $lan_1=lan_2$ gilt.

Der zweite Fall beinhaltet die Zuordnung einer Zeichenkette $b \in ALPH_{META}*$ als Bezeichner von zwei unterschiedlichen deskriptiven Symbolen in zwei unterschiedlichen natürlichen Sprachen $lan_1,lan_2 \in \{ger,fr,eng,...\}$ mit $lan_1 \neq lan_2$. In diesem Fall liegt eine *Homonymie i.w.S.* vor. Es kann nämlich sein, dass zwei unterschiedliche deskriptive Symbole $x_1,x_2 \in (K \cup OPS \cup RS)$ mittels wiederum zweier unterschiedlicher sprachspezifischer Bezeichnungsfunktionen bez_{lan_1} und bez_{lan_2} mit $lan_1 \neq lan_2$ auf dieselbe Zeichenkette $b \in ALPH*$ abgebildet werden. Die betroffene Zeichenkette b stellt in diesem Fall eine Überlappung der Bildmengen zweier sprachspezifischer Bezeichnungsfunktionen dar. Beispielsweise kann für die beiden Konzepte *Behaelter* und *Panzer* gelten: bez_{ger}(Behaelter)=`Tank` zw. bez_{eng}(Panzer)=`Tank`. Insofern stellt die Zeichenkette `Tank` ein Homonym dar.

In der Regel wird mit Homonymie lediglich die Eigenart einer Zeichenkette aus der *gleichen* Sprache erfasst. Mit den sprachspezifischen Bezeichnungsfunktionen lassen sich allerdings auch Homonymien von Zeichenketten aus unterschiedlichen Sprachen aufdecken.[2)] Dies ist dann der Fall, wenn zwei unterschiedliche deskriptive Symbole mittels zweier unterschiedlicher sprachspezifischer Bezeichnungsfunktionen bez_{lan_1}und bez_{lan_2} mit $lan_1 \neq lan_2$ auf die gleiche Zeichenkette $b \in ALPH*$ abgebildet werden. Zusätzlich verkompliziert wird dieser Umstand, wenn die sprachspezifischen Bezeichner bez-

1) Die Menge HOM aller homonymen Bezeichner wird im Folgenden in ihrer äquivalenten Charakterisierung als einstellige Relation angegeben, um eine Vergleichbarkeit mit der zweistelligen Relation SYN zur Kennzeichnung synonymer Bezeichner zu erhalten. So wird statt $b \in HOM$ die Schreibweise HOM(b) verwendet.

2) Zur Zulässigkeit der Auszeichnung sprachübergreifender Bezeichnerbeziehungen vgl. WEDEKIND ET AL. (2004), S. 340.

$_{lan_1}(x_1)$ und $bez_{lan_2}(x_2)$ zu zwei unterschiedlichen deskriptiven Symbolen x_1 und x_2 mit $x_1,x_2 \in (K \cup OPS \cup RS)$ in zwei unterschiedlichen natürlichen Sprachen lan_1 und lan_2 mit $lan_1 \neq lan_2$ „vertauscht" sind. Beispielsweise wird die Zeichenkette `Meer` im Holländischen dem Konzept als Bezeichner zugeordnet, dem im Deutschen der Bezeichner `See` zugeordnet ist. Umgekehrt wird im Holländischen der Bezeichner `See` dem Konzept zugeordnet, dem im Deutschen der Bezeichner `Meer` zugeordnet ist.[1)]

Synonyme sind voneinander unterschiedliche Zeichenketten, die gleichzeitig Bezeichner für dasselbe deskriptives Symbol $x \in (K \cup OPS \cup RS)$ sind.[2)] Sie werden in der dritten Spalte der Tabelle 9 vorgestellt. Im Fall von Synonymen wird ein deskriptives Symbol $x \in (K \cup OPS \cup RS)$ mit einer mehrelementigen Menge $bez_{lan}(x)$ von Zeichenketten bezeichnet (d.h. $|bez_{lan}(x)| \geq 2$). Die Bezeichner $b_1,...,b_n \in bez_{lan}(x)$ von x stehen in diesem Fall in einer Synonymbeziehung zueinander. So sind z.B. die Zeichenketten `C++` und `C Plus Plus` Synonyme, da beide Zeichenketten die Bezeichner *eines* Konzepts sind. Darüber hinaus können auch Zeichenketten aus den Bildern $bez_{lan_1}(x)$ und $bez_{lan_2}(x)$ zweier unterschiedlicher sprachspezifischer Bezeichnungsfunktionen bez_{lan_1} und bez_{lan_1} mit $lan_1 \neq lan_2$ zu einem deskriptiven Symbol $x \in (K \cup OPS \cup RS)$ als Synonyme betrachtet werden.[3)] Ebenso sind *Abkürzungen* von Zeichenketten als Synonyme zu den abgekürzten Zeichenketten anzusehen.

Das Bezeichnerpaar (b_1,b_2) ist Element der Synonymrelation

$$SYN \subseteq ALPH_{META}^* \times ALPH_{META}^*,$$

wenn b_1 und b_2 Synonyme sind. Es gilt für eine Sprache $lan \in \{ger,eng,fr,...\}$:

$$\forall b_1,b_2 \in ALPH_{META}^*: SYN(b_1,b_2) \leftrightarrow$$
$$\exists x \in (K \cup OPS \cup RS) : b_1 \in bez_{lan_1}(x) \wedge b_2 \in bez_{lan_2}(x).$$

Auch hierbei werden zwei Fälle unterschieden. Im ersten Fall stimmen die sprachspezifischen Bezeichnungsfunktionen bez_{lan_1} und bez_{lan_2} aufgrund von $lan_1=lan_2=lan$ überein. Dieser Fall beinhaltet die Zuordnung von zwei Zeichenketten $b_1,b_2 \in ALPH_{META}^*$ als sprachspezifische Bezeichner zu einem deskriptiven Symbol $x \in (K \cup OPS \cup RS)$ in einer natürlichen Sprache $lan_1=lan_2=lan$. In diesem Fall liegt eine *Synonymie i.e.S.* vor. Sämtliche Zeichenketten, die in der mehrelementigen Menge $bez_{lan}(x)$ enthalten sind, stehen somit zueinander in einer Synonymie-Beziehung i.e.S. zueinander. Der zweite Fall liegt vor, wenn aufgrund von $lan_1 \neq lan_2$ zwei unterschiedliche sprachspezifische

---

1) Vgl. HOPPENBROUWERS (1997), S. 42.

2) Vgl. LÖBNER (2003), S. 117; LUTZEIER (1995), S. 59 ff.; ORTNER (1997), S. 32.

3) Zu beachten sind dabei sprach- und kulturspezifische Umstände, die erst bei näherer Betrachtung auffallen. Hierunter fallen z.B. Zeichenketten, die vermeintlich in zwei unterschiedlichen Sprachen eine unterschiedliche Bedeutung haben, jedoch in mindestens einer der Sprachen ein Homonym darstellen; vgl. HOPPENBROUWERS (1997), S. 41 f. Thematisiert wird dies im Kontext der *sozialen Bedeutung* sprachlicher Konstrukte; vgl. LÖBNER (2003), S. 36 ff. Diese Problematik stellt sich hier allerdings nicht, da sowohl Konzepten als auch Operations- und Relationssymbolen sprachspezifische Bezeichner zugeordnet werden und nicht umgekehrt.

Bezeichnungsfunktionen bez_{lan_1} und bez_{lan_2} herangezogen werden. In diesem Fall liegt eine *Synonymie i.w.S.* vor. Dabei werden zwei Zeichenketten $b_1,b_2 \in ALPH_{META}*$ als Synonyme betrachtet, wenn beide das gleiche deskriptive Symbol $x \in (K \cup OPS \cup RS)$ in zwei unterschiedlichen natürlichen Sprachen lan_1 und lan_2 bezeichnen.

Die Familie deff der sprachspezifischen Definitionsfunktionen

$$def_{lan}=\{def_{ger}, def_{eng}, def_{fr},...\}$$

hat als Mitglieder Funktionen, mit denen deskriptiven Symbolen und Formeln über der ontologischen Signatur SIG_{OS} Definitionen in einer natürlichen Sprache[1)] zugeordnet werden können. Der Index $lan \in \{ger,eng,fr,...\}$ gibt – wie bei den Mitgliedern der Familie bezf auch – die jeweilige natürliche Sprache an, in der die Definition gilt. Wenn die natürliche Sprache in dem Argumentationskontext nicht von Relevanz ist, wird im Folgenden auch vereinfachend von einer sprachspezifischen Definitionsfunktion def_{lan} gesprochen.

Jede sprachspezifische Definitionsfunktion def_{lan} ist definiert als:

$$def_{lan}: (K \cup OPS \cup RS \cup FORM_{SIG_{OS}}) \rightarrow ALPH_{META}*.$$

Sprachspezifische Definitionsfunktionen können – im Gegensatz zu sprachspezifischen Bezeichnungsfunktionen – in ihren Argumenten auch ontologische Formeln aus der Menge $FORM_{SIG_{OS}}$ aller Formeln über der ontologischen Signatur SIG_{OS} aufnehmen. Dieses Ausdruckspotenzial sprachspezifischer Definitionsfunktionen erweist sich besonders dann als vorteilhaft, wenn zu Inferenz- und Integritätsregeln natürlichsprachliche Erläuterungen angegeben werden sollen.

Sprachspezifische Definitionsfunktionen sind grundsätzlich *partiell*. Es muss nämlich nicht jedem deskriptiven Symbol $x \in (K \cup OPS \cup RS)$ oder jeder ontologischen Formel $F \in FORM_{SIG_{OS}}$ eine sprachspezifische Definition $def_{lan}(x)$ bzw. $def_{lan}(F)$ zugewiesen werden. Ebenso sind sprachspezifische Definitionsfunktionen nicht notwendigerweise

---

1) Ein Alternativvorschlag zu *sprach*spezifischen Definitionsfunktionen findet sich in BERTOLAZZI ET AL. (2001), S. 5 f. Dort werden die Definitionen zu objektsprachlichen Konstrukten nicht *sprach*-, sondern *quellen*spezifisch ausgezeichnet. Je nachdem, welcher Quelle die Definition für das Konstrukt entstammt, müsste – bei Übertragung der dortigen Vorgehensweise auf ontologische Signaturen – eine für diese Quelle spezifische Definitionsfunktion def_{quelle} mit $quelle \in \{Quelle_1,..,Quelle_n\}$ herangezogen werden. Dieses Vorgehen hätte den Vorteil, dass die Quellenspezifität einer Definition auch immer eine Sprachspezifität implizit beinhalten würde, da jede Quelle eindeutig in einer natürlichen Sprache verfasst ist.

Allerdings wird von dieser Alternative aus mehreren Gründen abgesehen. Erstens kann nicht davon ausgegangen werden, dass für objektsprachliche Konstrukte, deren Definitionen notwendig sind, auch immer mindestens eine Quelle existiert, in der die Konstruktdefinition dokumentiert ist. Zweitens sollen mit Ontologien mentale Konstruktionsleitungen der Akteure abgebildet werden, die an der Ontologiekonstruktion beteiligt sind. Bei einer Einschränkung auf quellenspezifische Definitionen würden die mentalen Konstruktionsleistungen jener Akteure abgebildet werden, von denen die Quellen verfasst worden sind. Diese „Quellenverfasser“ brauchen mit den „Ontologiekonstrukteuren“ keineswegs übereinzustimmen.

injektiv. Es darf nämlich zwei unterschiedlichen Konstrukten die gleiche Definition zugewiesen werden. Beispielsweise können zwei äquivalente Formeln $F_1, F_2 \in FORM_{SIG_{OS}}$ mit $F_1 \neq F_2$ und $F_1 \equiv F_2$ durch die gleiche metasprachliche Zeichenkette $def_{lan}(F_1) = def_{lan}(F_2)$ definiert werden. Im Fall deskriptiver Symbole ist allerdings paarweise Unterschiedlichkeit der sprachspezifischen Definitionen zu erwarten, da mit jedem deskriptiven Symbol $x \in (K \cup OPS \cup RS)$ eine unterschiedliche Intension verbunden ist, die sich auch natürlichsprachlich einzigartig ausdrücken lässt.

Die Bildmenge einer sprachspezifischen Definitionsfunktion def_{lan} umfasst Zeichenketten über dem Alphabet $ALPH_{META}$. Im Unterschied zu einer Bezeichnungsfunktion bez_{lan} bildet eine Definitionsfunktionen def_{lan} deskriptive Symbole aus einer ontologischen Signatur SIG_{OS} und Formeln über SIG_{OS} nicht auf eine Teilmenge von $ALPH_{META}{}^*$, sondern auf ein Element von $ALPH_{META}{}^*$ ab. Dadurch wird den Elementen der Menge $(K \cup OPS \cup RS \cup FORM_{SIG_{OS}})$ in jeder Sprache lan höchstens *eine* Definition zugeordnet.

Die sprachspezifischen Definitionen von Formeln über ontologischen Signaturen sind natürlichsprachliche Erklärungen ihrer intensionalen Semantik. Sie erweisen sich dann als hilfreich, wenn eine ontologische Signatur SIG_{OS} um eine solche Menge von Formeln erweitert wird, die den Charakter von Anforderungen an die SIG_{OS}-Strukturen haben, mit denen die Ontologie extensional interpretiert werden kann. Jede Zeichenkette, die als sprachspezifische Definition einer ontologischen Formel $F \in FORM_{SIG_{OS}}$ zugeordnet ist, dient dazu, Benutzern einer Ontologie die Nachvollziehbarkeit der Anforderungskomponente zu vereinfachen. Beispielsweise kann die ontologische Formel

$F_1 =$ $\forall x_1,x_2,x_3,x_4$: (hat_Kompetenzdefizit(x_1,x_2) $\wedge$ hat_Kernkompetenz(x_1,x_3) $\wedge$
hat_Kompetenzdefizit(x_4,x_3) $\wedge$ hat_Kernkompetenz(x_4,x_2)) $\rightarrow$
geeignet_fuer_Kooperation(x_1,x_4)

durch die natürlichsprachliche Definition

$def_{ger}(F_1) =$ `Wenn die Kernkompetenzen zweier Unternehmen als Kompetenzdefizite des jeweils anderen Unternehmen angegeben sind, dann sind die beiden Unternehmen für eine Kooperation geeignet.`

erläutert werden.

Wenn zwei (syntaktisch) unterschiedliche Formeln F_1 und F_2 miteinander (semantisch) äquivalent sind ($F_1 \equiv F_2$), müssen ihre sprachspezifischen Definitionen $def_{lan}(F_1)$ bzw. $def_{lan}(F_2)$ nicht identisch sein, obwohl äquivalente Formeln immer die gleiche Bedeutung in allen SIG_{OS}-Strukturen haben. Beispielsweise kann die ontologische Formel

$F_1 = \forall x: A(x) \rightarrow B(x)$

durch

$def_{ger}(F_1)$=`Wenn x ein A ist, dann ist x auch ein B.`

erläutert werden. Die zu F_1 äquivalente Formel

$F_2=\forall x: \neg A(x) \vee B(x)$

kann hingegen erläutert werden durch

$def_{ger}(F_2)$=`Jedes x ist kein A oder ein B.`

Allerdings bietet es sich gerade für Formeln, die in eine äquivalente Subjugatsform transformierbar sind, an, die erstgenannte Variante zu definieren. Die „Wenn-Dann"-Formulierung einer natürlichsprachlichen Definition ist in der Regel einfacher nachzuvollziehen als die adjunktiv verknüpfte Variante.

Bei den Definitionen deskriptiver Symbole und Formeln handelt es sich um natürlichsprachliche Erläuterungen der Intension des jeweils definierten Konstruktes. Mit Hilfe der sprachspezifischen Definitionsfunktionen kann somit die intensionale Semantik eines deskriptiven Symbols oder einer Formel natürlichsprachlich angegeben werden. Im Gegensatz zu der extensionalen Semantik ist allerdings die intensionale Semantik, die durch sprachspezifische Definitionen festgelegt wird, nicht *formal* bestimmt. Während die extensionale Semantik deskriptiver Symbole durch Bestimmung der Mengen aus Objekten und Objekttupeln angegeben wird, die Instanzen des deskriptiven Symbols sind, wird die intensionale Semantik durch natürlichsprachliche Zeichenketten angegeben.[1)] Somit handelt es sich bei der intensionalen Semantik, die durch sprachspezifische Definitionen festgelegt wird, um *informale* Semantiken.

Der Zusammenhang zwischen intensionaler und extensionaler Semantik in ontologischen Signaturen wird in der Abbildung 12 verdeutlicht:

---

1) Grundsätzlich ist auch die formale Analyse der Zeichenketten vorstellbar, mit denen deskriptive Symbole definiert werden. Hierfür kommen insbesondere spracherkennende Systeme in Frage; vgl. CARSTENSEN ET AL. (2001), S. 469 ff. Die formale Analyse der natürlichsprachlichen Definitionen wird sogar dadurch begünstigt, dass sie induktiv über dem Alphabet $ALPH_{META}$ konstruiert sind. Die formale Analyse der natürlichsprachlichen Zeichenketten über dem Alphabet $ALPH_{META}$ liegt allerdings außerhalb des Erkenntnisinteresses der vorliegenden Arbeit.

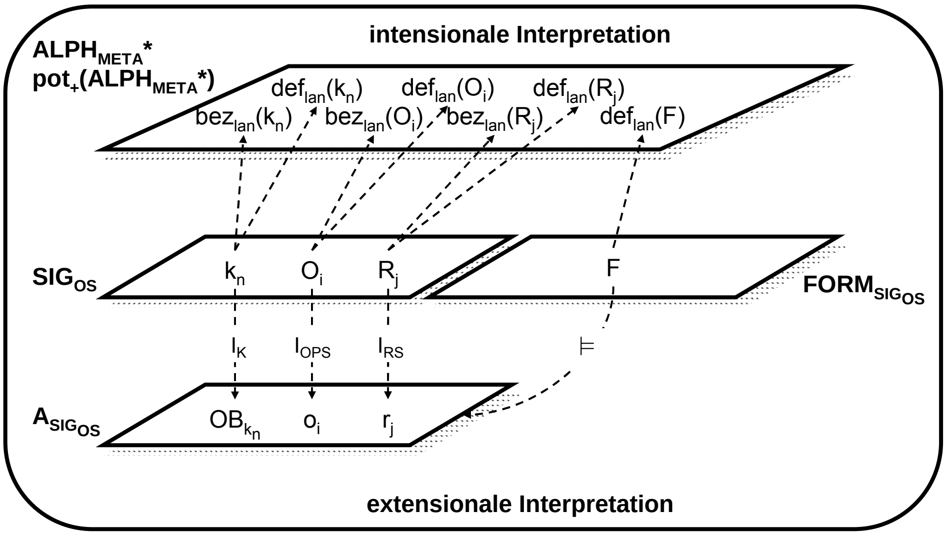

Abbildung 12: intensionale und extensionale Semantiken ontologischer Signaturen

In der Abbildung 12 wird die Separation zwischen formal- und natürlichsprachlichen Konstrukten aufgegriffen und illustriert. Zentrales Element der Abbildung ist die mittlere Schicht, die sich aus zwei Bereichen zusammensetzt: Der linke Bereich der mittleren Schicht umfasst die deskriptiven Symbole aus einer ontologischen Signatur SIG_{OS}. Es handelt sich dabei um Konzepte sowie um Operations- und Relationssymbole. Der rechte Bereich der mittleren Schicht umfasst Formeln, die mit den Symbolen aus SIG_{OS} konstruiert werden können. Die unterste Schicht repräsentiert eine SIG_{OS}-Struktur $A_{SIG_{OS}}$, mit der die ontologische Signatur SIG_{OS}-Struktur extensional interpretiert werden kann.

Die oberste Schicht der Abbildung 12 umfasst die Mengen $ALPH_{META}$* und $pot_{+}(ALPH_{META}*)$ aller Zeichenketten bzw. Mengen von Zeichenketten über dem Alphabet $ALPH_{META}$. Mit den Pfeilen, die von der mittleren Schicht zu der oberen Schicht führen, werden die sprachspezifischen Bezeichnungs- und Definitionsfunktionen angedeutet. Einerseits ordnen die Pfeile den deskriptiven Symbolen aus SIG_{OS} natürlichsprachliche Zeichenketten als Bezeichner zu. Andererseits werden mit den Pfeilen deskriptiven Symbolen aus SIG_{OS} und Formeln über SIG_{OS} natürlichsprachliche Zeichenketten als Definitionen zugeordnet.

Die Pfeile, die von der mittleren linken Schicht zu der unteren Schicht führen, verdeutlichen Interpretationsfunktionen, mit denen den deskriptiven Symbolen aus der ontologischen Signatur SIG_{OS} jeweils Konstrukte aus einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ zugewiesen werden. Bei den zugewiesenen Konstrukten handelt es sich jeweils um die Extension desjenigen Symbols aus dem Pfeilursprung. So wird beispielsweise das Konzept k_n aus der mittleren Schicht durch die konzeptspezifische Objektmenge OB_{k_n} aus der unteren Schicht extensional interpretiert. Mit dem Pfeil von der rechten mittleren zur unteren Schicht wird hingegen die Modellbeziehung zwischen einer ontologischen Formel F über SIG_{OS} und der SIG_{OS}-Struktur $A_{SIG_{OS}}$ illustriert. Sie ist genau dann gegeben, wenn die SIG_{OS}-Struktur $A_{SIG_{OS}}$ derart aufgebaut ist, dass die Formel F in $A_{SIG_{OS}}$ gültig ist.

Die Abbildung 12 weist eine Verwandtschaft mit dem (*semiotischen*) *Bedeutungsdreieck* auf, mit dem des Öfteren[1)] die Zusammenhänge zwischen Konzepten[2)], Bezeichnern und den Extensionen von Konzepten verdeutlicht wird. Es geht in seiner ursprünglichen Form auf ARISTOTELES zurück,[3)] wurde allerdings in der Variante, die für sprachanalytische Zwecke von Relevanz ist, vornehmlich von OGDEN/RICHARDS verwendet.[4)] Mit dem Bedeutungsdreieck wird in der Regel der Zusammenhang zwischen einem mentalen Konzept als Intension (1. Ecke), der Extension des mentalen Konzepts (2. Ecke) und der Bezeichnung des mentalen Konzepts (3. Ecke) illustriert. Übertragen auf sprachliche Konzepte aus ontologischen Signaturen, deren Extensionen und deren Bezeichnungen kann das „traditionelle“ Bedeutungsdreieck entsprechend Abbildung 13 zu einem *erweiterten Bedeutungsdreieck* ausgebaut werden.

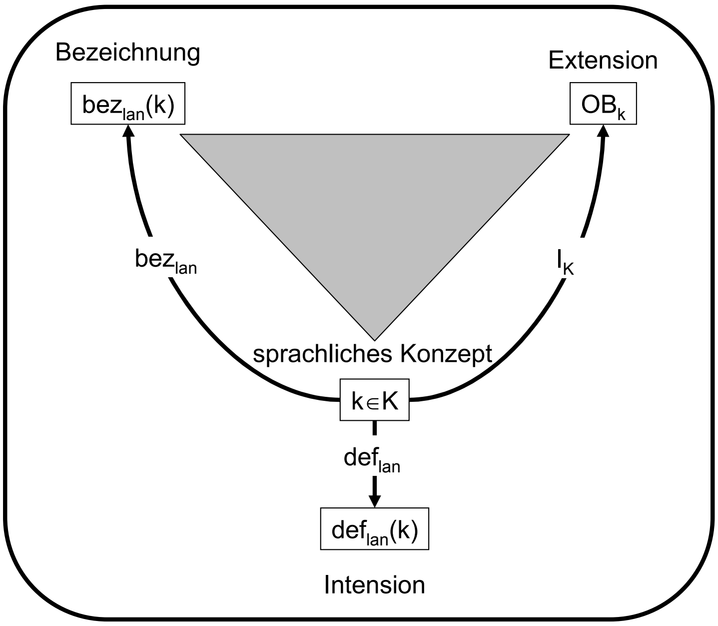

Abbildung 13: Erweitertes Bedeutungsdreieck

Üblicherweise wird der untere Teil der Abbildung 13 ausgelassen, da die informale Definition von Konzepten nicht Betrachtungsgegenstand ist. Außerdem werden die Beziehungen zwischen den Ecken des Dreiecks zumeist nur durch natürlichsprachliche Be-

1) Vgl. BOMAN ET AL. (1997), S. 27; HANSEN ET AL. (1992), S. 8; HARRAS (2000), S. 13.; HOPPENBROUWERS (1997), S. 28; LÖBNER (2003), S. 32; MÄDCHE (2002), S. 14; PFLÜGLMAYER (2001), S. 131; SOWA (2000), S. 191 ff.; SURE (2003), S. 27. Vgl. darüber hinaus SCHEFE (1999), S. 123, zu einer Übertragung des Bedeutungsdreiecks auf softwaretechnische Aspekte.

2) In der Regel werden exemplarisch aufgeführte Bedeutungsdreiecke nur auf Konzepte angewandt. Die Illustration lässt sich allerdings ohne weiteres auch auf Operations- und Relationssymbole ausweiten.

3) Vgl. ECO (1988), S. 192 f.

4) Vgl. OGDEN/RICHARDS (1972), S. 11 ff.

schriftungen illustriert.[1)] In der erweiterten Form spiegelt das Bedeutungsdreieck die Form wider, die für die Semantik ontologischer Signaturen festgelegt ist. Demnach wird jedem sprachlichen Konzept $k \in K$ durch eine sprachspezifische Bezeichnungsfunktion bez_{lan} eine Menge $bez_{lan}(k)$ von Zeichenketten als Bezeichner zugeordnet. Die Extension $I_K(k)$ jedes Konzepts k wird durch die konzeptspezifische Objektmenge $I_K(k)=OB_k$ angegeben, mit der das jeweilige Konzept extensional interpretiert wird.

3.1.4 Pragmatische Aspekte von Ontologien

3.1.4.1 Ontologische Spezifikationen

Analog zum Ausbau konventioneller und sortierter Signaturen zu prädikatenlogischen Spezifikationen werden ontologische Signaturen zu *ontologischen Spezifikationen* ausgebaut. Der Ausbau umfasst eine Menge ontologischer Formeln, die den Charakter von Anforderungen an SIG_{OS}-Strukturen haben, mit denen die ontologischen Signaturen intensional interpretiert werden können. Die Menge $A(SIG_{OS})$ aller SIG_{OS}-Strukturen zu einer ontologischen Signatur SIG_{OS} umfasst nämlich eine Vielzahl von Strukturen, die bei Kenntnis der intensionalen Semantik der objektsprachlichen Konstrukte aus SIG_{OS} ausgeschlossen werden können. Durch die Hinzunahme von Formeln über einer ontologischen Signatur SIG_{OS}, deren Gültigkeit notwendig ist, kommen weniger SIG_{OS}-Strukturen als Modelle der Formelmenge in Frage.

Einerseits fallen hierunter solche Strukturen, die ausgeschlossen werden können, weil sie die „unzulässige“ extensionale Interpretation von SIG_{OS} ermöglichen würden. Um die Unzulässigkeit von SIG_{OS}-Strukturen kenntlich machen zu können, müssen Anforderungen formuliert werden, denen alle zulässigen SIG_{OS}-Strukturen zu genügen haben. Anforderungen bezüglich der Zulässigkeit von SIG_{OS}-Strukturen werden in Ontologien mit Hilfe von *Integritätsregeln* formuliert.

Andererseits gilt es auch solche Strukturen für die extensionale Interpretation auszuschließen, in denen konkrete Beziehungen zwischen Individuen fehlen, die bei Kenntnis der intensionalen Semantik der objektsprachlichen Konstrukte aus SIG_{OS} gegeben sein müssten. Sind in einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ Beziehungen nicht enthalten, die aufgrund des „Sinns“ deskriptiver Symbole darin enthalten sein müssten, so ist in $A_{SIG_{OS}}$ implizites Wissen enthalten, das es zu erschließen gilt. Die Erschließung impliziten Wissens erfolgt in ontologischen Spezifikationen mit Hilfe von *Inferenzregeln*.

Die Differenzierung zwischen den beiden Regelarten – Integritäts- und Inferenzregeln – erfolgt aufgrund der *Zweckorientierungen*, die mit ihnen verbunden sind. Integritätsregeln werden zu Zwecken der Bestimmung der *Zulässigkeit* wissensrepräsentierender Formelsysteme spezifiziert. Inferenzregeln werden hingegen zu Zwecken der *Erschließung* neuer Formeln aus bereits vorhandenen wissensrepräsentierenden Formeln spezi-

1) Vgl. hierzu die zuvor aufgeführten Quellen zum Bedeutungsdreieck.

fiziert. Daher weisen ontologische Spezifikationen neben syntaktischen und semantischen Aspekten auch *pragmatische* Aspekte[1)] auf.

Ontologische Spezifikationen umfassen – wie prädikatenlogische Spezifikationen auch – eine Anzahl von Regeln, die über der zugrunde liegenden Signatur SIG_{OS} konstruiert werden. Während allerdings prädikatenlogische Spezifikationen lediglich Anforderungen *eines* bestimmten Typs zulassen, wird in ontologischen Spezifikationen zwischen zwei voneinander unterschiedlichen Anforderungsarten unterschieden. Es handelt sich hierbei um die eingangs erwähnten Inferenz- und Integritätsregeln. Entsprechend ist eine ontologische Spezifikation nicht – wie im Fall prädikatenlogischer Spezifikationen – als Zwei-, sondern als Drei-Tupel definiert:

$$SPEZ_{OS}=(SIG_{OS},INF_{SIG_{OS}},INT_{SIG_{OS}}).$$

Die Komponenten einer ontologischen Spezifikation $SPEZ_{OS}$ sind:

(1.) eine ontologische Signatur SIG_{OS},

(2.) eine Menge $INF_{SIG_{OS}}$ von *objektsprachlichen*[2)] *Inferenzregeln* mit $INF_{SIG_{OS}} \subset FORM_{SIG_{OS}}$ und

(3.) eine Menge $INT_{SIG_{OS}}$ von *objektsprachlichen*[3)] *Integritätsregeln* mit $INT_{SIG_{OS}} \subset FORM_{SIG_{OS}}$ mit

(4.) $(INF_{SIG_{OS}} \cup INT_{SIG_{OS}}) \not\subseteq KODM_{SIG_{OS}}$ und

(5.) $INF_{SIG_{OS}} \cap INT_{SIG_{OS}} = \varnothing$.

---

1) Im Rahmen der *syntaktischen* Analyse sprachlicher Konstrukte werden die Beziehungen zwischen den Zeichen untereinander untersucht, mit denen die sprachlichen Konstrukte konstruiert sind. Gegenstand der *semantischen* Dimension ist die Beziehung zwischen einerseits dem sprachlichen und andererseits dem bezeichneten Konstrukt. Der *Benutzer* des sprachlichen Konstruktes wird dabei weder im Rahmen der syntaktischen noch im Rahmen der semantischen Analyse berücksichtigt. Er findet erst im Rahmen der *pragmatischen* Analyse eine Berücksichtigung. Hierbei werden insbesondere die Zwecksetzungen untersucht, mit denen Benutzer sprachliche Konstrukte verwenden.

2) Der Zusatz „objektsprachliche" ist im Normalfall notwendig, um die hier gemeinten Inferenzregeln von metasprachlichen Inferenzregeln eines prädikatenlogischen Kalküls unterscheiden zu können. Als „Inferenzregeln" der Prädikatenlogik werden üblicherweise solche metasprachlichen Ausdrücke bezeichnet, die die Ableitung gültiger objektsprachlicher Ausdrücke (Konklusions-Formeln) aus einer Menge gültiger objektsprachlicher Ausdrücke (Antezedenz-Formeln) erlauben. Somit nehmen metasprachliche Inferenzregeln in ihren Argumenten objektsprachliche Ausdrücke auf. Zu den „bekanntesten" metasprachlichen Inferenzregeln der Prädikatenlogik gehören der *modus ponens* und der *modus tollens*. Sie wurde in früheren Anmerkungen bereits vorgestellt. Wenn nicht explizit hervorgehoben, sind im Folgenden stets objektsprachliche Inferenzregeln gemeint.

3) Analog zu Inferenzregeln wird im Folgenden der Zusatz „objektsprachliche" im Kontext von Integritätsregeln nicht verwendet. Im Fall von Integritätsregeln ist von keiner Verwechslungsgefahr mit metasprachlichen Konstrukten auszugehen, da die Bezeichnung „metasprachliche Integritätsregel" im Rahmen der Prädikatenlogik nicht geläufig ist.

Eine ontologische Spezifikation $SPEZ_{OS}$ ist die Erweiterung einer ontologischen Signatur SIG_{OS} um eine Menge $INF_{SIG_{OS}}$ von *Inferenzregeln* und eine Menge $INT_{SIG_{OS}}$ von *Integritätsregeln*. Bei beiden Regelarten handelt es sich jeweils um eine echte Teilmenge der Menge $FORM_{SIG_{OS}}$ aller Formeln über der ontologischen Signatur SIG_{OS}. Zudem werden an die Spezifikation von Inferenz- und Integritätsregeln durch die Definitionskomponenten 4.) und 5.) Anforderungen formuliert. Zum einen darf die Menge $(INF_{SIG_{OS}} \cup INT_{SIG_{OS}})$ aller Inferenz- oder Integritätsregeln nicht widersprüchlich sein. Zum zweiten darf keine Formel existieren, die sowohl als Inferenz- als auch als Integritätsregel spezifiziert ist.

In der Definition von ontologischen Spezifikationen ist keine Forderung enthalten, dass die Mengen $INF_{SIG_{OS}}$ oder $INT_{SIG_{OS}}$ nicht-leer sein müssen. Wenn ihre Mengen $INF_{SIG_{OS}}$ *und* $INT_{SIG_{OS}}$ leer sind, degeneriert eine ontologische Spezifikation $SPEZ_{OS}=(SIG_{OS}, \emptyset,\emptyset)$ zu einer ontologischen Signatur SIG_{OS}. Daher kann jede ontologische Signatur SIG_{OS} zu einer ontologischen Spezifikation $SPEZ_{OS}=(SIG_{OS},\emptyset,\emptyset)$ ausgebaut werden.[1] Um eine begriffliche Differenzierung zwischen ontologischen Spezifikationen mit mindestens einer nicht-leeren Menge $INF_{SIG_{OS}}$ oder $INT_{SIG_{OS}}$ und ontologischen Spezifikationen, die „verlustfrei“ auf ontologische Signaturen zurückgeführt werden können, zu ermöglichen, werden im Folgenden als *ontologische Spezifikationen i.w.S.* solche ontologischen Spezifikationen bezeichnet, die mindestens eine nicht-leere Menge $INF_{SIG_{OS}}$ oder $INT_{SIG_{OS}}$ haben. Bei ontologischen Signaturen handelt es sich entsprechend um *ontologische Spezifikationen i.e.S.*, da in ontologischen Signaturen grundsätzlich weder Inferenz- noch Integritätsregeln vorgesehen sind.

Die Bezeichnungen „Ontologie“ und „ontologische Spezifikation“ werden im Folgenden synonym verwendet. Somit werden als „Ontologie i.w.S.“ alle ontologischen Spezifikationen i.w.S. bezeichnet. In einem enger gefassten Begriffsverständnis umfasst der Begriff der *Ontologie i.e.S.* alle ontologischen Spezifikationen i.e.S. Daher handelt es sich bei allen ontologischen Signaturen um Ontologien i.e.S. Falls nicht explizit in der Argumentation hervorgehoben, wird im weiteren Verlauf stets von einer Ontologie i.w.S. ausgegangen, wenn der Zusatz entfällt.

Sowohl bei den Elementen der Menge $INF_{SIG_{OS}}$ als auch bei den Elementen der Menge $INT_{SIG_{OS}}$ handelt es sich um *Klauseln*, deren Quantoren explizit aufgeführt werden. Klauseln sind solche Formeln, die entweder genau ein *Literal* lit oder die Adjunktion

1) Ontologien werden teilweise sogar auf eine ontologische Signatur beschränkt; vgl. BENCH-CAPON ET AL. (2003), S. 705; BENCH-CAPON/MALCOLM (1999), S. 254. Zu einer analogen Definition prädikatenlogischer Signaturen als prädikatenlogische Spezifikationen vgl. EHRIG ET AL. (1999), S. 175 („Jede Signatur ist auch eine Signatur-Spezifikation, ...“) und EHRICH ET AL. (1989), S. 181 („ ... jede algebraische Signatur auch eine algebraische Spezifikation ist.“). Es wird bei dieser Argumentation davon abgesehen, dass eine ontologische Spezifikation $SPEZ_{OS}=(SIG_{OS},\emptyset,\emptyset)$ zumindest die Option freihält, Elemente ihrer Mengen $INF_{SIG_{OS}}$ und $INT_{SIG_{OS}}$ zu spezifizieren.

($lit_1 \vee ... \vee lit_n$) von n Literalen mit $n \geq 2$ und $n \in \mathbb{N}_+$ sind.[1] Jedes Literal lit ist wiederum entweder eine atomare ontologische Formel $R_j(t_1,...,t_n)$ oder die Negation $\neg R_j(t_1,...,t_n)$ einer atomaren ontologischen Formel. Im ersten Fall wird das Literal $lit=R_j(t_1,...,t_n)$ als *positiv* bezeichnet. Im zweiten Fall ist $lit=(\neg R_j(t_1,...,t_n))$ ein *negatives* Literal. Somit nehmen Inferenz- und Integritätsregeln folgende Form an:

$$\forall x_1,...,x_q: (\neg R_1(t_{1_1},...,t_{1_{l(1)}})) \vee ... \vee (\neg R_n(t_{n_1},...,t_{n_{l(n)}})) \vee$$
$$(R_{n+1}(t_{n+1_1},...,t_{n+1_{l(n+1)}})) \vee ... \vee (R_{n+m}(t_{n+m_1},...,t_{n+m_{l(n+m)}})).$$

In den Argumenten der Literale aus einer Regel können Variablen vorkommen. Für jede Variable, die in einer Inferenz- oder Integritätsregel vorkommt, wird üblicherweise ihre *implizite Allquantifizierung* angenommen. In diesen Fällen wird beispielsweise statt der ontologischen Formel $(\forall x: \neg R_1(x) \vee R_2(x))$ die Formel $(\neg R_1(x) \vee R_2(x))$ verwendet. Der Verfasser bevorzugt hingegen die Spezifikation von Inferenz- und Integritätsregeln, in der die Allquantifizierung von Variablen stets explizit angegeben ist. Dadurch wird zum einen die technische Handhabbarkeit von Inferenz- und Integritätsregeln vereinfacht. Zum anderen werden dadurch syntaktische „Brüche" zwischen Inferenz- und Integritätsregeln einerseits und wissensrepräsentierenden Formeln aus einer ontologiegestützten Wissensbasis andererseits vermieden.

In der Regel handelt es sich bei Inferenz- und Integritätsregeln um *subjunktive Klauseln*. Subjunktive Klauseln sind solche Klauseln, in denen mindestens ein negatives Literal $lit=R_1(t_{1_1},...,t_{1_{l(1)}})$ und mindestens ein positives Literal $lit=(\neg R_2(t_{2_1},...,t_{1_{l(2)}})$ vorkommt.[2] Mit Hilfe der prädikatenlogischen Äquivalenzen lassen sich subjunktive Klauseln zu *Subjugatsformeln* umformen. Zu diesem Zweck werden aus einer Klausel kl mit:

$$kl=\forall x_1,...,x_k: lit_1 \vee ... \vee lit_n \vee lit_{n+1} \vee ... \vee lit_{n+m}$$

alle negativen Literale

$$lit_1=\neg R_1(t_{1_1},...,t_{1_{l(1)}}) ,..., lit_n=\neg R_n(t_{n_1},...,t_{n_{l(n)}})$$

als positive Literale konjunktiv miteinander links vom Subjugatspfeil $\rightarrow$ verknüpft und alle positiven Literale

$$lit_{n+1}=R_{n+1}(t_{n+1_1},...,t_{1_{l(1)}}) ,..., lit_{n+m}=R_{n+1+m}(t_{n+m_1},...,t_{n+m_{l(n+m)}})$$

adjunktiv miteinander rechts vom Subjugatspfeil $\rightarrow$ verknüpft. Die o.a. Formel kann dann als:

$$F_1 = \forall x_1,...,x_q: (R_1(t_{1_1},...,t_{1_{l(1)}}) \wedge ... \wedge R_n(t_{n_1},...,t_{n_{l(n)}})) \rightarrow$$
$$(R_{n+1}(t_{n+1_1},...,t_{n+1_{l(n+1)}})) \vee ... \vee (R_{n+m}(t_{n+m_1},...,t_{n+m_{l(n+m)}}))$$

1) Vgl. BEIERLE/KERN-ISBERNER (2000), S. 59 f.; BIBEL (1992), S. 28; DAHR (1994), S. 8 f.; EBBINGHAUS ET AL. (1992), S. 246; EHRIG ET AL. (1999), S. 291 ff.; LAUTENBACH (2003), S. 278; SCHÖNING (1992), S. 39.

2) Vgl. ZELEWSKI (1995), Bd. 5.1, S. 172.

angegeben werden.[1)] In Anlehnung an die Darstellung als Subjugatsformel können die Literale, die in der ursprünglich adjunktiven Darstellung negativ waren und in der subjunktiven Darstellung links vom Subjugatspfeil „→“ als positive Literale platziert sind, als *Antezedenzformeln* bezeichnet werden. Literale, die in der ursprünglich adjunktiven Darstellung positiv waren und in der subjunktiven Darstellung weiterhin positiv rechts vom Subjugatspfeil „→“ platziert sind, können als *Konklusionsformeln* bezeichnet werden.

Klauseln können hinsichtlich der Anzahl ihrer positiven und negativen Literale unterschieden werden.[2)] *Horn-Klauseln* sind solche Klauseln, in denen höchstens ein positives Literal vorkommt.[3)] In ihrer subjunktiven Darstellung zeichnen sich Horn-Klauseln dadurch aus, dass sie höchstens ein positives Literal als Konklusionsformel aufweisen und zudem alle vorkommenden Antezedenzformeln positive Literale sind. Die o.a. Formel F_1 ist bei $n \geq 2$ keine Horn-Klausel, da in der adjunktiven Darstellung mindestens zwei positive Literale vorkommen. Die Klausel

$$F_2 = \forall x_1,\ldots,x_q: \neg R_1(t_{1_1},\ldots,t_{1_{l(1)}}) \vee \ldots \vee \neg R_n(t_{n_1},\ldots,t_{n_{l(n)}}) \vee R_{n+1}(t_{n+1_1},\ldots,t_{n+1_{l(n+1)}})$$

ist hingegen eine Horn-Klausel. Sie kann in ihrer subjunktiven Form als

$$F_2 \equiv \forall x_1,\ldots,x_q: (R_1(t_{1_1},\ldots,t_{1_{l(1)}}) \wedge \ldots \wedge R_n(t_{n_1},\ldots,t_{n_{l(n)}})) \rightarrow R_{n+1}(t_{n+1_1},\ldots,t_{n+1_{l(n+1)}})$$

angegeben werden.

Obwohl die meisten Werkzeuge für Ontologien lediglich zur Verarbeitung von Inferenz- und Integritätsregeln in der Lage sind, die als Horn-Klauseln vorliegen, wird diese Einschränkung für das integrative Modellierungskonzept nicht vorgenommen. Insofern erweist sich das Modellierungskonzept als ausdruckmächtiger als Konzepte, die auf Horn-Klauseln beschränkt sind. Für die algorithmische Transformation von Inferenz- und Integritätstransitionen werden lediglich einige *syntaktische Anforderungen* an Inferenz- und Integritätsregeln gestellt. Demnach hat für alle Inferenz- und Integritätsregeln zu gelten, dass sie

(1.) in ihrer subjunktiven Form vorliegen,

---

1) Die Umformung der Formel

$$F_1 = \forall x_1,\ldots,x_q: \neg R_1(t_{1_1},\ldots,t_{1_{l(1)}}) \vee \ldots \vee \neg R_n(t_{n_1},\ldots,t_{n_{l(n)}}) \vee R_{n+1}(t_{n+1_1},\ldots,t_{n+1_{l(1)}}) \vee \ldots \vee R_{n+m}(t_{n+m_1},\ldots,t_{n+m_{l(n+m)}})$$

erfolgt in folgenden Schritten:

$$F_1 \equiv \forall x_1,\ldots,x_q: \neg(R_1(t_{1_1},\ldots,t_{1_{l(1)}}) \wedge \ldots \wedge R_n(t_{n_1},\ldots,t_{n_{l(n)}})) \vee (R_{n+1}(t_{n+1_1},\ldots,t_{n+1_{l(1)}}) \vee \ldots \vee R_{n+m}(t_{n+m_1},\ldots,t_{n+m_{l(n+m)}}))$$

$$\equiv \forall x_1,\ldots,x_q: (R_1(t_{1_1},\ldots,t_{1_{l(1)}}) \wedge \ldots \wedge R_n(t_{n_1},\ldots,t_{n_{l(n)}})) \rightarrow (R_{n+1}(t_{n+1_1},\ldots,t_{n+1_{l(1)}}) \vee \ldots \vee (R_{n+m}(t_{n+m_1},\ldots,t_{n+m_{l(n+m)}}))$$

2) Vgl. DAHR (1994), S. 9.

3) Vgl. DAS (1992), S. 105 ff. („definite clauses“); GALLIER (1986), S. 411 f.; KREOWSKI (1991), S. 94; SCHMID/ KINDSMÜLLER (1996) S. 96; SCHÖNING (1992), S. 32 f.

(2.) alle entweder positiven oder negativen Literale aus dem Antezedenz konjunktiv verknüpft sind und

(3.) sie höchstens ein Literal als Konklusionsformel aufweisen.[1)]

Als Inferenz- oder Integritätsregel zugelassen wäre beispielsweise:

$$\forall x_1,\ldots,x_q: (lit_1 \wedge\ldots\wedge lit_n) \rightarrow lit_{n+1}.$$

Horn-Klauseln erfüllen in ihrer subjunktiven Darstellung die Anforderung an Inferenz- und Integritätsregeln, denn jede Horn-Klausel weist in ihrer subjunktiven Darstellung höchstens ein positives Literal in ihrer Konklusionskomponente auf. Darüber hinaus sind in Horn-Klauseln auch in ihrer Antezedenz-Komponente nur positive Literale enthalten. Dennoch sind die syntaktischen Anforderungen an Inferenz- und Integritätsregeln für die Ausdrucksmächtigkeit von Ontologien keineswegs einschränkender Natur, da jede Klausel kl sich entsprechend den syntaktischen Anforderungen an Inferenz- und Integritätsregeln umformen lässt.

Die o.a. Formel F_1 kann beispielsweise[2)] zu

$$F_1 \equiv \forall x_1,\ldots,x_q: (R_1(t_{1_1},\ldots,t_{1_{l(1)}}) \wedge\ldots\wedge R_n(t_{n_1},\ldots,t_{n_{l(n)}}) \wedge \\ \neg R_{n+1}(t_{n+1_1},\ldots,t_{n+1_{l(n+1)}}) \wedge\ldots\wedge \neg R_{n+m-1}(t_{n+m-1_1},\ldots,t_{n+m-1_{l(n+m)}})) \rightarrow R_{n+m}(t_{n+m_1},\ldots,t_{n+m_{l(n+m)}})$$

äquivalent umgeformt werden.[3)] In dieser Form, wäre die Formel als Inferenz- oder Integritätsregel zugelassen.

---

1) Vgl. BEIERLE/KERN-ISBERNER (2000), S. 69.

2) Um den syntaktischen Anforderungen an Inferenz- und Integritätsregeln gerecht werden zu können, bieten sich für die o.a. Formel F_1 mehrere Möglichkeiten an. Es kommt beispielsweise jede Formel in Frage, bei der eine der Konklusionsformeln aus der Konklusionskomponente entfernt und in ihrer negierten Form dem Antezedenzteil der Formel konjunktiv angeschlossen wird.

3) Die erste Umformung der Formel

$$F_1 = \forall x_1,\ldots,x_q: \neg R_1(t_{1_1},\ldots,t_{1_{l(1)}}) \vee \ldots \vee \neg R_n(t_{n_1},\ldots,t_{n_{l(n)}}) \vee R_{n+1}(t_{n+1_1},\ldots,t_{n+1_{l(1)}}) \vee \ldots \vee \\ R_{n+m}(t_{n+m_1},\ldots,t_{n+m_{l(n+m)}})$$

zu einer Inferenz- oder Integritätsregel erfolgt nach folgendem Schema:

$$F_1 \equiv \forall x_1,\ldots,x_q: \neg(R_1(t_{1_1},\ldots,t_{1_{l(1)}}) \wedge \ldots \wedge (R_n(t_{n_1},\ldots,t_{n_{l(n)}})) \vee \\ (R_{n+1}(t_{n+1_1},\ldots,t_{n+1_{l(1)}}) \vee \ldots \vee R_{n+m}(t_{n+m_1},\ldots,t_{n+m_{l(n+m)}}))$$

$$\equiv \forall x_1,\ldots,x_q: \neg(R_1(t_{1_1},\ldots,t_{1_{l(1)}}) \wedge \ldots \wedge R_n(t_{n_1},\ldots,t_{n_{l(n)}}) \wedge \neg R_{n+1}(t_{n+1_1},\ldots,t_{n+1_{l(1)}}) \wedge \ldots \wedge \\ \neg R_{n+m-1}(t_{n+m-1_1},\ldots,t_{n+m-1_{l(n+m-1)}})) \vee R_{n+m}(t_{n+m_1},\ldots,t_{n+m_{l(n+m)}}))$$

$$\equiv \forall x_1,\ldots,x_q: (R_1(t_{1_1},\ldots,t_{1_{l(1)}}) \wedge \ldots \wedge R_n(t_{n_1},\ldots,t_{n_{l(n)}}) \wedge \neg R_{n+1}(t_{n+1_1},\ldots,t_{n+1_{l(1)}}) \wedge \ldots \wedge \neg R_{n+m-1}(t_{n+m-1_1},\ldots,t_{n+m-1_{l(n+m)}})) \rightarrow R_{n+m}(t_{n+m_1},\ldots,t_{n+m_{l(n+m)}})$$

Äquivalent dazu ist auch die Formel

$$F_1 \equiv \forall x_1,...,x_q: (R_1(t_{1_1},...,t_{1_{l(1)}}) \wedge ... \wedge R_{n-1}(t_{n-1_1},...,t_{n-1_{l(n-1)}}) \wedge \neg R_{n+1}(t_{n+1_1},...,t_{n+1_{l(n+1)}}) \wedge ... \wedge \neg R_{n+m}(t_{n+m_1},...,t_{n+m_{l(n+m)}})) \rightarrow \neg R_n(t_{n_1},...,t_{n_{l(n)}})$$

als Inferenz- oder Integritätsregel zugelassen.[1)]

Zu beachten ist, dass den Elementen der Mengen $INF_{SIG_{OS}}$ und $INT_{SIG_{OS}}$ mit Hilfe der sprachspezifischen Definitionsfunktionen def_{lan} natürlichsprachliche Definitionen zugewiesen werden können. Im Argumentbereich jeder sprachspezifischen Definitionsfunktion def_{lan} werden nämlich alle Elemente der $FORM_{SIG_{OS}}$ und somit auch alle Elemente von $INF_{SIG_{OS}}$ und $INT_{SIG_{OS}}$ zugelassen.[2)] Mit Hilfe einer sprachspezifischen Definitionsfunktion def_{lan} können somit alle Formeln, die über einer ontologischen Signatur SIG_{OS} konstruiert sind, auf jeweils eine natürlichsprachliche Zeichenkette $zk \in ALPH_{META}*$ abgebildet werden. Dieses Ausdruckspotenzial von Ontologien erweist sich insbesondere dann als vorteilhaft, wenn Regeln einen Komplexitätsgrad haben, durch den sie sich einer menschlichen Verarbeitung entziehen. Die induktive Grammatik zur Konstruktion ontologischer Formeln erlaubt nämlich die Konstruktion von Formeln beliebiger Komplexität. Die sprachspezifischen Definitionsfunktionen wurden insbesondere in ontologische Signaturen einbezogen, damit auch menschliche Akteure, die u.U. keine prädikatenlogischen Vorkenntnisse aufweisen, die „Bedeutung" von Inferenz- und Integritätsregeln erfassen können.

Die extensionale Semantik einer Ontologie $SPEZ_{OS}=(SIG_{OS},INF_{SIG_{OS}},INT_{SIG_{OS}})$ wird über der Menge $A(SIG_{OS})$ aller SIG_{OS}-Strukturen zu der ontologischen Signatur SIG_{OS} bestimmt, die $SPEZ_{OS}$ zugrunde liegt. Alle SIG_{OS}-Strukturen aus der Menge $A(SIG_{OS})$ können nämlich für die extensionale Interpretation der ontologischen Signatur SIG_{OS}

---

1) Die zweite Umformung der Formel

$$F_1 = \forall x_1,...,x_q: \neg R_1(t_{1_1},...,t_{1_{l(1)}}) \vee ... \vee \neg R_n(t_{n_1},...,t_{n_{l(n)}}) \vee R_{n+1}(t_{n+1_1},...,t_{n+1_{l(1)}}) \vee ... \vee R_{n+m}(t_{n+m_1},...,t_{n+m_{l(n+m)}})$$

zu einer Inferenz- oder Integritätsregel erfolgt nach folgendem Schema:

$$F_1 \equiv \forall x_1,...,x_q: \neg R_1(t_{1_1},...,t_{1_{l(1)}}) \vee ... \vee \neg R_{n-1}(t_{n-1_1},...,t_{n-1_{l(n-1)}})) \vee \neg R_n(t_{n_1},...,t_{n_{l(n)}}) \vee R_{n+1}(t_{n+1_1},...,t_{n+1_{l(1)}}) \vee ... \vee R_{n+m}(t_{n+m_1},...,t_{n+m_{l(n+m)}})$$

$$\equiv \forall x_1,...,x_q: \neg(R_1(t_{1_1},...,t_{1_{l(1)}}) \wedge ... \wedge R_{n-1}(t_{n-1_1},...,t_{n-1_{l(n-1)}}) \wedge \neg R_{n+1}(t_{n+1_1},...,t_{n+1_{l(1)}}) \wedge ... \wedge \neg R_{n+m}(t_{n+m_1},...,t_{n+m_{l(n+m)}})) \vee \neg R_n(t_{n_1},...,t_{n_{l(n)}}))$$

$$\equiv \forall x_1,...,x_q: (R_1(t_{1_1},...,t_{1_{l(1)}}) \wedge ... \wedge R_{n-1}(t_{n-1_1},...,t_{n-1_{l(n-1)}}) \wedge \neg R_{n+1}(t_{n+1_1},...,t_{n+1_{l(1)}}) \wedge ... \wedge \neg R_{n+m-1}(t_{n+m-1_1},...,t_{n+m-1_{l(n+m)}})) \rightarrow \neg R_n(t_{n_1},...,t_{n_{l(n)}})$$

2) Der Zielbereich jeder sprachspezifischen Definitionsfunktion def_{lan} enthält hingegen nur Zeichenketten über dem natürlichsprachlichen Alphabet $ALPH_{META}$.

herangezogen werden.[1] Im Gegensatz zu ontologischen Signaturen erfahren jedoch ontologische Spezifikationen eine zweifache Erweiterung. Die erste Erweiterung betrifft die Menge $INF_{SIG_{OS}}$ aller Inferenzregeln, die analog zu der Menge $REG_{SIG_{PL}}$ aller Regeln in einer prädikatenlogischen Spezifikation $SPEZ_{PL}$ definiert sind. Mit Hilfe der Inferenzregeln werden aus der Menge $A(SIG_{OS})$ jene SIG_{OS}-Strukturen „aussortiert", die zwar zu der zugrunde gelegten ontologischen Signatur SIG_{OS} passen, allerdings aufgrund ihrer zu „knappen" Beschaffenheit für die extensionale Interpretation der Ontologie $SPEZ_{OS}$ ausgeschlossen werden können. Der Ausschluss von SIG_{OS}-Strukturen ist dadurch bedingt, dass sie nicht den Anforderungen genügen, die durch Inferenzregeln formuliert sind. Dabei nehmen Inferenzregeln Bezug auf die *Gültigkeit* von Formeln über der ontologischen Signatur SIG_{OS}. Komponenten der zweiten Erweiterung $INT_{SIG_{OS}}$ beziehen sich hingegen auf die *Zulässigkeit* von Formeln. Auf beide Erweiterungen wird im Folgenden näher eingegangen.

Inferenz- und Integritätsregeln sind zwei wesentliche Komponenten von Ontologien, die trotz ihrer Bedeutung oftmals in der Literatur vernachlässigt werden. Dies liegt in erster Linie daran, dass sowohl Inferenz- als auch Integritätsregeln nur von wenigen Sprachen zur Konstruktion von Ontologien unterstützt werden.[2] Für den Ansatz der vorliegenden Arbeit sind Inferenz- und Integritätsregeln aus drei Gründen von besonderem Interesse.

Erstens wird mit Inferenz- und Integritätsregeln die wesentliche *Ausdrucksökonomie* von Ontologien erschlossen. Insbesondere Inferenzregeln erlauben nämlich die Erschließung gültiger ontologischer Formeln aus einer anderen Menge gültiger ontologischer Formeln. Somit brauchen nicht alle wissensrepräsentierenden Formeln explizit spezifiziert zu werden. Es reicht aus, solche Formeln zu spezifizieren, die in Kombination mit Inferenzregeln die Erschließung von zusätzlichen wissensrepräsentierenden Formeln erlauben. Mit Integritätsregeln können dagegen Aussagen im Vorfeld ihrer Gültigkeitsprüfung ausgeschlossen werden, wenn sie unzulässig sind. Analog zu dem Übergang von der konventionellen zur sortierten Prädikatenlogik, der u.a. dadurch motiviert war, bereits in ihrer Grammatik solche objektsprachlichen Zeichenketten als Aussagen ausschließen zu können, die keinen „Sinn" haben, werden durch Integritätsregeln Aussagen ausgeschlossen, wenn sie aufgrund realweltlicher Umstände unzulässig sind.

Zweitens sind sowohl Inferenz- als auch Integritätsregeln Ausdrucksmittel, die im Rahmen der Unternehmensmodellierung auf breite Resonanz stoßen.[3] Oftmals lassen

---

1) Diese Annahme gilt nur unter der Prämisse, dass jede SIG_{OS}-Struktur $A_{SIG_{OS}}$ aus der Menge $A(SIG_{OS})$ in dem Sinne „passend" zu der ontologischen Signatur SIG_{OS} ist, als dass $A_{SIG_{OS}}$ für jedes objektsprachliche Konstrukt aus SIG_{OS} mindestens ein Konstrukt umfasst, mit dem SIG_{OS} extensional interpretiert werden kann. Gewährleistet wird dies durch die bijektiven Interpretationsfunktionen aus der Familie I_{OS}.

2) Vgl. Abschnitt 5.1.1.

3) Vgl. ERWIN (2002), S. 143 ff.; FRANK/VAN LAAK (2003), S. 83; KNOLMAYER ET AL. (2000), S. 18 ff.; RAM/KHATRI (2005), S. 89 ff.; ROSCA ET AL. (2002), S. 363 ff.

sich nämlich regelhafte Zusammenhänge in Unternehmen mit Hilfe von Inferenz- oder Integritätsregeln auf kompakte Weise ausdrücken.

Besonders interessant sind hierbei Inferenz- und Integritätsregeln im Kontext der *Referenzmodellierung*.[1)] Es handelt sich hierbei um eine bewusst – vordergründig zu Zwecken ihrer Wiederverwendbarkeit – abstrakt gehaltene Modellierung einer Domäne. Referenzmodelle werden oftmals als Vorlage für die Unternehmensmodellierung verwendet. Daher wird Referenzmodellen auch öfters ein *normativer* Charakter zugesprochen.[2)] Werden Inferenz- und Integritätsregeln in Referenzmodellen berücksichtigt, so haben auch die Regeln einen normativen Charakter. Dabei könnte durch die abstrakte Beschreibung von regelhaften Zusammenhängen die Wiederverwendbarkeit von Referenzmodellen erhöht werden. Beispielsweise kann ein regelhafter Zusammenhang, der für eine Branche typisch ist, in Form einer Inferenzregel in einem ontologiegestützten Referenzmodell spezifiziert werden. Die branchenspezifisch gegebene Unzulässigkeit von Zuständen kann analog in Integritätsregeln spezifiziert werden.

Darüber hinaus erweisen sich sowohl Inferenz- als auch Integritätsregeln für das integrative Modellierungskonzept der vorliegenden Arbeit als äußerst fruchtbar. Sie lassen sich nämlich mit verschiedenen Transitionsarten, die aus höheren Petri-Netzen bekannt sind, in Einklang bringen. Auf diesen letzten Aspekt wird im Kontext der Integration von Ontologien und Petri-Netzen ausführlich eingegangen.[3)]

Um die Unterschiede zwischen Inferenz- und Integritätsregeln präzise ausmachen zu können, wird ein *semiotischer* Bezugsrahmen entfaltet. Der semiotische Bezugsrahmen ist insofern geeignet, die Unterschiede zwischen Inferenz- und Integritätsregeln auszumachen, als dass es sich bei beiden um *sprachliche* Konstrukte handelt. Für die Analyse sprachlicher Konstrukte werden im semiotischen Bezugsrahmen die Dimensionen der *Syntax*, der *Semantik* und der *Pragmatik* herangezogen.

Hinsichtlich der syntaktischen Dimension können zwischen Inferenz- und Integritätsregeln keine Unterschiede ausgemacht werden. Ausprägungen beider Regelarten werden nämlich als Formeln über einer ontologischen Signatur SIG_{OS} konstruiert. Somit liegt beiden Regelarten die induktive Grammatik zur Konstruktion ontologischer Formeln zugrunde. Zwar existieren hiervon abweichende Auffassungen, nach denen die Notation von Integritätsregeln eine andere sein müsse als die von Inferenzregeln,[4)] allerdings

1) Zu Referenzmodellen vgl. SCHÜTTE (1998), S. 69 ff.; SCHWEGMANN (1999), S. 53 ff., und die Beiträge in BECKER ET AL. (1999).

2) Zum normativen Charakter von Referenzmodellen vgl. FETTKE/LOOS (2004), S. 333.

3) Vgl. Abschnitt 4.2.2.1.2

4) Vgl. DAS (1992), S. 275. Dort werden Inferenzregeln in Anlehnung an die Schreibweise in der logischen Programmierung in der Form

$$A \leftarrow B$$

notiert. Integritätsregeln werden hingegen in der üblichen Form als wird hiervon in der vorliegenden Arbeit abgesehen. Die alternative Notation wird nämlich lediglich dadurch motiviert, Integritätsregeln auch als solche kenntlich machen zu können. Hierdurch würde allerdings bereits auf die *pragmatische* Dimension von Inferenz- und Integritätsregeln vorgegriffen werden. Dadurch käme es zu einer nicht wünschenswerten Vermengung der syntaktischen und pragmatischen Dimension.[1)] Um Integritätsregeln als solche kenntlich zu machen, wird in ontologischen Spezifikationen eine Sektion $INT_{SIG_{OS}}$ aufgeführt, die alle Integritätsregeln einer ontologischen Signatur umfasst.

Ebenso sind in der Semantik von Inferenz- und Integritätsregeln keine Unterschiede auszumachen. Ausprägungen beider Regelarten stehen nämlich genau dann in der Modellbeziehung $\models$ zu einer SIG_{OS}-Struktur $A_{SIG_{OS}}$, wenn sie darin gültig sind. Dies ist der wesentliche Baustein der formalen Semantik, die Inferenz- und Integritätsregeln zugrunde liegt. Sowohl Inferenz- als auch Integritätsregeln können nämlich lediglich hinsichtlich ihrer *Form* in einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ ausgewertet werden.

Obwohl für Ausprägungen beider Regelarten das Auswertungsverfahren identisch definiert ist, wird es lediglich für Inferenzregeln in Anspruch genommen. Die formale Semantik einer Ontologie $SPEZ_{OS}$ wird nämlich nur über der Menge $INF_{SIG_{OS}}$ aller Inferenzregeln definiert. Integritätsregeln werden hierbei nicht berücksichtigt. An diesem Punkt wird bereits an die *pragmatische* Dimension von Inferenz- und Integritätsregeln angeknüpft. Inferenzregeln werden nämlich von einem Ontologiekonstrukteur zu dem *Zweck* konstruiert, aus einer Menge gültiger Formeln solche Formeln abzuleiten, die bei gleichzeitiger Gültigkeit der Inferenzregel auch gültig sein müssen. In diesem Sinn können Inferenzregeln als die *konstruktive* Regelkomponente einer Ontologie $SPEZ_{OS}$ charakterisiert werden. Ihre Konstruktivität ist auf ihr Potenzial zur *Faktenerzeugung* zurückzuführen. Integritätsregeln lassen sich hingegen als die *destruktive* Regelkomponente einer Ontologie $SPEZ_{OS}$ charakterisieren. Im Gegensatz zu Inferenzregeln dienen Integritätsregeln nicht dazu, Fakten zu erzeugen. Mit ihrer Hilfe werden lediglich Zulässigkeiten von wissensrepräsentierenden Formeln überprüft. Entsprechend werden Formeln, die im Widerspruch zu den Anforderungen einer Integritätsregel stehen, als unzulässig bezüglich dieser Regel bezeichnet.

Der konstruktive Charakter von Inferenzregeln wird anhand eines Beispiels verdeutlicht. Wenn beispielsweise eine ontologische Formel

$$F_1 = R_1(t_1,t_2)$$
$$\text{mit } A_{SIG_{OS}} \models F_1$$

und eine Inferenzregel

$$A \rightarrow B$$

notiert. Der inhaltliche Unterschied zwischen Inferenz- und Integritätsregeln wird allerdings auch bei DAS nicht ausreichend thematisiert.

1) Vgl. GODFREY ET AL. (1998), S. 267.

$$F_2 = \forall x_1,x_2: R_1(x_1,x_2) \rightarrow R_2(x_2,x_1)$$
$$\text{mit } A_{SIG_{OS}} \models F_2$$

vorliegen, dann kann darauf geschlossen werden, dass auch für die Formel

$$F_3=R_2(t_2,t_1)$$

gelten muss:

$$A_{SIG_{OS}} \models F_3.$$

Die Gültigkeit der Formel F_3 in $A_{SIG_{OS}}$ ist nämlich bei Gültigkeit der Inferenzregel F_2 in $A_{SIG_{OS}}$ in Verbindung mit der Gültigkeit der Formel F_1 in $A_{SIG_{OS}}$ notwendig. Entsprechend der Inferenzregel F_2 muss bei Gültigkeit einer Formel $R_1(t_1,t_2)$ in $A_{SIG_{OS}}$ auch die Formel $R_2(t_2,t_1)$ in $A_{SIG_{OS}}$ gültig sein. Bei Substitution der Variablen x_1 und x_2 in der Inferenzregel F_2 durch die Terme t_1 bzw. t_2 hat entsprechend die Formel $R_2(t_2,t_1)$ in $A_{SIG_{OS}}$ gültig zu sein. Wäre die Formel F_3 in $A_{SIG_{OS}}$ ungültig, so müsste auch ($A_{SIG_{OS}} \not\models F_1$) oder ($A_{SIG_{OS}} \not\models F_2$) gelten. Da allerdings sowohl die Gültigkeit von F_1 als auch die Gültigkeit von F_2 vorausgesetzt wurden, hat auch F_3 gültig zu sein.[1)]

Objektsprachliche Inferenzregeln lassen sich demnach tendenziell dazu nutzen, von der Gültigkeit einer Formelmenge $FM_1 \subseteq FORM_{SIG_{OS}}$ auf die Gültigkeit einer weiteren Formelmenge $FM_2 \subseteq FORM_{SIG_{OS}}$ zu schließen. Das Verfahren der Erschließung von Wissen bezüglich der Gültigkeit der Formelmenge FM_2 aus dem Wissen der Gültigkeiten der Menge $INF_{SIG_{OS}}$ und der Formelmenge FM_1 wird als *Inferenz* bezeichnet.[2)]

---

1) Die Ableitbarkeit der ontologischen Formel F_3 aus der Formelmenge $F_1 \cup F_2$ ist auf die metasprachliche Inferenzregel *modus ponens* zurückführbar. Entsprechend dem modus ponens kann nämlich auf die Gültigkeit einer Formel B geschlossen werden, wenn die Gültigkeit der Formelmenge {A, A→B} bekannt ist. Dabei wird der modus ponens unter der Verwendung der Ableitungsrelation $\vdash$ wie folgt notiert:

$$\{A, A \rightarrow B\} \vdash B.$$

Im umgekehrten Fall des *modus tollens* kann hingegen auf die Gültigkeit einer Formel $\neg A$ geschlossen werden, wenn die Gültigkeit der Formeln $\neg B$ und $A \rightarrow B$ bekannt sind. Unter Verwendung der Ableitungsrelation $\vdash$ kann der modus tollens wie folgt notiert werden:

$$\{\neg B, A \rightarrow B\} \vdash \neg B.$$

2) Voraussetzung hierfür ist die *Vollständigkeit* und *Korrektheit* des zugrunde gelegten Inferenzkalküls. Die Vollständigkeit des Inferenzkalküls ist genau dann gegeben, wenn alle Formeln, die gültig sein müssen, weil andere Formeln gültig sind, mittels Inferenzregelanwendungen abgeleitet werden können. Die Korrektheit des Inferenzkalküls ist genau dann gegeben, wenn aus gültigen Formeln auch nur gültige Formeln abgeleitet werden können. Bei Korrektheit und Vollständigkeit des Inferenzkalküls gilt die Folgerungsrelation $\Vdash$ zwischen Formelmengen genau dann, wenn die Ableitungsrelation $\vdash$ zwischen ihnen erfüllt ist. Auf diesen Aspekt wurde in Abschnitt 3.1.3.1.2.2 hingewiesen. Wegen der vorausgesetzten Vollständigkeit und Korrektheit des Inferenzkalküls für die Ableitungsrelation $\vdash$ wird im Folgenden weiterhin die bereits eingeführte Folgerungsrelation $\Vdash$ verwendet, und zwar auch dann, wenn syntaktisch definierte Ableitungen mittels Inferenzregeln betrachtet werden.

Alle ontologischen Formeln $F_1,...,F_n \in FM_2$, die aus einer Formelmenge FM_1 und einer Menge $INF_{SIG_{OS}}$ von Inferenzregeln inferenziell erschlossen werden können, werden als *Theoreme* der Formelmenge ($FM_1 \cup INF_{SIG_{OS}}$) bezeichnet. Im Fall einer Formelmenge $FM \subseteq FORM_{SIG_{OS}}$ umfasst die Menge

$$TM(FM)=\{ F \mid (FM \Vdash F)\}$$

alle Theoreme von FM. Zu beachten ist dabei, dass in der Menge TM(FM) aller Theoreme einer Formelmenge FM sowohl alle Formeln $F_1,...,F_n \in FM$ einzeln als auch in miteinander konjugierter Form vorkommen. Darüber hinaus umfasst die Theoremmenge TM(FM) auch alle Formeln, die zwar nicht in der Formelmenge FM enthalten sind, aber von FM abgeleitet werden können, da sie bei Gültigkeit von FM auch gültig sein müssen.

Die Theoreme einer Formelmenge $FM \cup INF_{SIG_{OS}}$ sind demnach definiert als:

$$TM(FM \cup INF_{SIG_{OS}})= \{F \mid (FM \cup INF_{SIG_{OS}} \Vdash F)\}.$$

Die Menge $TM(FM \cup INF_{SIG_{OS}})$ umfasst alle Theoreme, die aus einer Menge FM von ontologischen Formeln und einer Menge $INF_{SIG_{OS}}$ von Inferenzregeln inferenziell erschlossen werden können. Wenn $TM(FM \cup INF_{SIG_{OS}}) = (FM \cup INF_{SIG_{OS}})$ gilt, ist die Formelmenge $(FM \cup INF_{SIG_{OS}})$ eine *Theorie*. In diesem Fall können mit Hilfe der Inferenzregeln aus $INF_{SIG_{OS}}$ aus FM keine „neuen“ Formeln erschlossen werden, die nicht in $INF_{SIG_{OS}}$ oder FM – zumindest implizit – enthalten sind.

Mit Inferenzregeln können u.a. Eigenschaften von Operations- und Relationssymbolen formuliert werden. Durch die Angabe der Eigenschaften der Symbole kann ihre intensionale Semantik weiter präszisiert werden. Es handelt sich hierbei in erster Linie um Eigenschaften, mit denen die Menge OPS aller Operationssymbole und die Menge RS aller Relationssymbole *geordnet* werden können. Analog zu der partiellen Ordnung, die durch die Subkonzeptrelation $\sqsubseteq$ auf der Menge K aller Konzepte definiert ist, können Unterordnungsbeziehungen zwischen ontologischen Operations- und Relationssymbolen mit Hilfe von Inferenzregeln spezifiziert werden. Die Ordnungsbeziehung wird allerdings im Gegensatz zu der partiellen Ordnung auf der Menge K nicht durch eine *meta*sprachliche Relation zum Ausdruck gebracht, sondern durch *objekt*sprachliche Inferenz- und Integritätsregeln.1)

1) Grundsätzlich bestünde auch hier die Möglichkeit zur Einführung metasprachlicher Relationen, um Unterordnungsbeziehungen zwischen Operations- und Relationssymbolen spezifizieren zu können. Allerdings steht die dadurch erzeugte Komplexität in ungünstigem Verhältnis zu dem hinzugewonnenen Ausdruckskomfort. Von den vielfachen Problemen, die im Umfeld metasprachlicher Relationen für Operations- und Relationssymbole zu beachten sind, wird im Folgenden lediglich auf das Problem der Stelligkeit eingegangen (für weitere Aspekte vgl. KANEIWA (2004), S. 162 ff.).

Wenn z.B. gefordert werden soll, dass die Formel $R_2(x_1,x_2)$ gültig sein muss, wenn die Formel $R_1(x_1,x_2)$ gültig ist, dann wird das durch die Inferenzregel

$$\forall x_1,x_2: R_1(x_1,x_2) \rightarrow R_2(x_1,x_2)$$

ausgedrückt. Mit der Formel wird eine Unterordnung des Relationssymbols R_1 gegenüber dem Relationssymbol R_2 ausgedrückt. Wenn die Formel $R_1(x_1,x_2)$ von einer Familie $belf_{OS}$ konzeptspezifischer Variablenbelegungen in einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ bestätigt wird, dann muss bei der gleichen Variablenbelegung $belf_{OS}$ auch die Formel $R_2(x_1,x_2)$ in $A_{SIG_{OS}}$ bestätigt werden, wenn die o.a. Inferenzregel in $A_{SIG_{OS}}$ gültig ist. Der umgekehrte Fall muss allerdings nicht gelten. So kann bei Gültigkeit der o.a. Formel die Teilformel $R_2(x_1,x_2)$ bestätigt werden, ohne dass $R_1(x_1,x_2)$ bestätigt werden muss. Die Bestätigung von $R_2(x_1,x_2)$ ist nämlich für die Bestätigung von $R_1(x_1,x_2)$ zwar notwendig, aber nicht hinreichend. Umgekehrt ist die Bestätigung von $R_1(x_1,x_2)$ für die Bestätigung von $R_2(x_1,x_2)$ zwar nicht notwendig, aber hinreichend.

Die hierarchische Anordnung von Relationssymbolen kann in analoger Weise auch für Operationssymbole formuliert werden. Um allerdings Unterordnungsbeziehungen zwischen Operationssymbolen formulieren zu können, bedarf es mindestens eines Relationssymbols. Es wird benötigt, um die ontologischen (Teil-) Formeln im Antezedenz und in der Konklusion einer ordnenden Inferenzregel ausdrücken zu können. Wenn z.B. die Unterordnung eines Operationssymbols O_1 gegenüber einem Operationssymbol O_2 ausgedrückt werden soll, kann folgende Regel hierfür verwendet werden:

Für Unterordnungsbeziehungen zwischen Operations- und Relationssymbolen aus einer ontologischen Signatur SIG_{OS} untereinander müssten stets auch deren Stelligkeiten berücksichtigt werden. Wenn ein Relationssymbol R_1 mit $typ_{RS_{OS}}(R_1)=(k_{1_1}...k_{1x})$ einem Relationssymbol R_2 mit $typ_{RS_{OS}}(R_2)=(k_{2_1}...k_{2y})$ mit x=y untergeordnet wird, bereitet das weniger Probleme. Durch die Unterordnung würde in diesem Fall ausgedrückt werden, dass bei Gültigkeit einer atomaren Formel $R_1(t_1,...,t_n)$ auch die atomare Formel $R_2(t_1,...,t_n)$ gültig zu sein hat. Weitaus schwieriger ist hingegen der Fall mit $y \neq x$. Für den Fall, dass z.B. ein Relationssymbol R_1 mit $typ_{RS_{OS}}(R_1)=(k_1k_2)$ einem ontologischen Relationssymbol R_2 mit der Typisierung $typ_{RS_{OS}}(R_2)=(k_3k_4k_5)$ untergeordnet werden soll, bereitet es zwar keine Probleme, dass keines der Konzepte k_1 und k_2 in der Typisierung von R_2 vorkommt. Die Konzepte k_3,k_4 und k_5 in der Typisierung von R_2 könnten nämlich ihrerseits als Superkonzepte von k_1 und k_2 spezifiziert werden. Vielmehr ist die *Länge* der Konzeptkette problematisch, mit der R_2 typisiert ist. Üblicherweise ist nämlich mit dem Übergang auf ein übergeordnetes Ausdrucksmittel ein Detailverlust verbunden. Dieser Detailverlust äußert sich z.B. darin, dass, wenn das Konzept *Mann* dem Konzept *Mensch* mittels $\sqsubseteq$ untergeordnet wird, die Intension von *Mensch* von der Intension von *Mann* umfasst wird. Die Eigenart, ein Mensch zu sein, ist zwar notwendig, aber nicht hinreichend für die Eigenart, ein Mann zu sein. Hinreichende Bedingungen für die Eigenart, ein Mann zu sein, gehen bei der Beschreibung eines Mannes lediglich durch das Konzept *Mensch* teilweise verloren. Analog würden auch bei dem Übergang von einem untergeordneten Relationssymbol R_1 zu einem übergeordneten Relationssymbol R_2 Details verloren gehen; vgl. KANEIWA (2004), S. 160. Wenn allerdings das in Bezug auf R_1 übergeordnete Relationssymbol R_2 mehr Stellen aufweist als R_1, werden zusätzliche Verfahren benötigt, um zu bestimmen, welche Terme an den zusätzlichen Stellen vorkommen müssen.

$\forall x_1,x_2$: Equal($O_1(x_1),x_2$) $\rightarrow$ Equal($O_2(x_1),x_2$)

Mit der Unter- und Überordnung von Operations- und Relationssymbolen wird eine Teilmengenbeziehung zwischen den jeweiligen Extensionen ausgedrückt. Wenn z.B. das Relationssymbol $R_1 \in RS$ dem Relationssymbol $R_2 \in RS$ durch eine Inferenzregel untergeordnet wird, bedeutet dies, dass die Extension $I_{RS}(R_1)=r_1$ eine Teilmenge der Extension $I_{RS}(R_2)=r_2$ sein muss. Sämtliche Zwei-Tupel (ob_1,ob_2), die in der Relation $r_1 \in RF$ enthalten sind, müssen auch in der Relation $r_2 \in RF_{OS}$ enthalten sein. Somit gilt der Zusammenhang:

$$(A_{SIG_{OS}} \models (\forall x_1,x_2: R_1(x_1,x_2) \rightarrow R_2(x_1,x_2))) \rightarrow (I_{RS}(R_1) \subseteq I_{RS}(R_2)).$$

Die Spezifikation einer Ordnungsrelation auf den Mengen OPS und RS aller Operations- bzw. Relationssymbole aus einer ontologischen Signatur SIG_{OS} durch Inferenzregeln ist ein Instrument zu der *vertikalen* Anordnung der Symbole. Daneben gibt es weitere Möglichkeiten, mit denen eine *horizontale* Anordnung durchgeführt werden kann. Hierzu gehört z.B. die Spezifikation von *symmetrischen*, *transitiven* und *inversen* Operations- und Relationssymbolen. Beispielsweise kann mit Hilfe der Inferenzregel

$\forall x_1,x_2$: $R_1(x_1,x_2) \leftrightarrow R_1(x_2,x_1)$

die Symmetrie eines Relationssymbols R_1 ausgedrückt werden. Hierfür kommt beispielsweise das Relationssymbol *Verheiratet_mit* mit intuitiver intensionaler Semantik in Frage. Beispielsweise muss genau dann, wenn die Formel $R_1(t_1,t_2)$ und die o.a. Inferenzregel in einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ gültig sind, auch die Formel $R_2(t_2,t_1)$ in $A_{SIG_{OS}}$ gültig sein.

Die Transitivität eines Relationssymbols R_1 kann hingegen mit Hilfe der Inferenzregel

$\forall x_1,x_2,x_3$: $R_1(x_1,x_2) \wedge R_1(x_2,x_3) \rightarrow R_1(x_1,x_3)$

ausgedrückt werden. Ein Beispiel hierfür ist das Relationssymbol *Vorgesetzter_von*. Schließlich kann die inverse Beziehung zwischen zwei Relationssymbolen R_1 und R_2 wie folgt ausgedrückt werden:

$\forall x_1,x_2$: $R_1(x_1,x_2) \leftrightarrow R_2(x_2,x_1)$

Beispielsweise stehen die Relationssymbole *Arbeitet_fuer* und *Hat_Mitarbeiter* in einer inversen Beziehung zueinander.

Im Gegensatz zu Inferenzregeln werden Integritätsregeln nicht dazu verwendet, aus einer gültigen Formelmenge wiederum gültige Formeln zu erschließen. Integritätsregeln werden vielmehr zu dem Zweck konstruiert, die *Zulässigkeit* von Formelmengen zu überprüfen.[1)] Die Gültigkeit einer ontologischen Formel $F \in FORM_{SIG_{OS}}$ ist nämlich stets in Bezug auf eine SIG_{OS}-Struktur $A_{SIG_{OS}}$ definiert. Eine ontologische Formel F ist dem-

1) Vgl. GODFREY ET AL. (1998), S. 269.

nach genau dann in einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ gültig, wenn sie von jeder Familie $belf_{OS}$ konzeptspezifischer Variablenbelegungen in $A_{SIG_{OS}}$ bestätigt wird. Analog ist eine Formelmenge $FM \subseteq FORM_{SIG_{OS}}$ genau dann in einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ gültig, wenn jede Formel $F_x \in FM$ in $A_{SIG_{OS}}$ gültig ist.

Mit dem Begriff *Zulässigkeit* wird lediglich mittelbar Bezug auf die Gültigkeit von Formelmengen genommen. Der Begriff Zulässigkeit ist nicht – wie der Begriff *Gültigkeit* – in Bezug auf eine SIG_{OS}-Struktur $A_{SIG_{OS}}$ bestimmt, sondern in Bezug auf eine Integritätsregel. Die Zulässigkeit einer Formelmenge FM in Bezug auf eine Integritätsregel $F \in INT_{SIG_{OS}}$ ist genau dann gegeben, wenn die Gültigkeit von F in einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ die Gültigkeit von FM in $A_{SIG_{OS}}$ nicht ausschließt. Somit nehmen Integritätsregeln lediglich unmittelbar Bezug auf die Gültigkeit von Formeln und Formelmengen. Mit ihnen werden bestimmte Extensionen zu objektsprachlichen Konstrukten – die grundsätzlich möglich wären – ausgeschlossen.[1)]

Der Begriff *Zulässigkeit* kann mit den Hilfsmitteln, die im Rahmen der formalen Semantik ontologischer Formeln vorgestellt wurden, weiter präzisiert werden. Es handelt sich hierbei insbesondere um die Menge $KODM_{SIG_{OS}}$ widersprüchlicher Formelmengen über einer ontologischen Signatur SIG_{OS}. Die Menge $KODM_{SIG_{OS}}$ umfasst alle Mengen ontologischer Formeln über SIG_{OS}, deren Elemente in keiner SIG_{OS}-Struktur $A_{SIG_{OS}}$ gemeinsam gültig sein können. Der Begriff Zulässigkeit wird somit unmittelbar an den Begriff *Widersprüchlichkeit* gekoppelt. Eine ontologische Formel $F_1 \in FORM_{SIG_{OS}}$ ist nämlich genau dann bezüglich einer Integritätsregel $F_2 \in INT_{SIG_{OS}}$ zulässig, wenn die Formelmenge $FM=(F_1 \cup F_2)$ nicht widersprüchlich ist. Es muss also mindestens eine SIG_{OS}-Struktur $A_{SIG_{OS}}$ geben, in der die Formelmenge $FM=(F_1 \cup F_2)$ erfüllbar ist. Wenn das der Fall ist, gilt $FM \notin KODM_{SIG_{OS}}$.

Die Definition der Zulässigkeit von Formeln kann auf Formelmengen ausgeweitet werden. Demnach ist eine Formelmenge $FM_1 \subseteq FORM_{SIG_{OS}}$ genau dann bezüglich einer Integritätsregel $F \in INT_{SIG_{OS}}$ zulässig, wenn die Formelmenge $FM=(FM_1 \cup F)$ nicht widersprüchlich ist und somit $(FM_1 \cup FM_2) \notin KODM_{SIG_{OS}}$ gilt. Analog lässt sich Argumentation auf Mengen von Integritätsregeln ausweiten. Eine Formelmenge $FM_1 \subseteq FORM_{SIG_{OS}}$ ist genau dann bezüglich einer Menge $FM_2 \subseteq INT_{SIG_{OS}}$ von Integritätsregeln zulässig, wenn die Formelmenge $FM_1 \cup FM_2$ nicht widersprüchlich ist. Dabei hat allerdings stets zu gelten, dass die Menge $INT_{SIG_{OS}}$ selbst nicht widersprüchlich sein darf, da sonst die Un-

1) Solche integritätsbewahrenden Mechanismen sind für Ontologien bereits mit Hilfe anderer Komponenten vorgestellt worden. Beispielsweise stellen die Typisierungsfunktionen $typ_{OPS_{OS}}$ und $typ_{RS_{OS}}$ Ausdruckmittel dar, mit denen als Extensionen zu Operations- bzw. Relationssymbolen nur solche Operationen bzw. Relationen zugelassen werden, die mit der Typisierung des jeweiligen Symbols verträglich sind. Im Unterschied zu den bereits bekannten Mitteln zur Bewahrung der Integrität in SIG_{OS}-Strukturen sind jedoch Integritätsregeln aus einer ontologischen Spezifikation $SPEZ_{OS}$ grundsätzlich *objektsprachlicher* Natur. Es handelt sich nämlich bei ihnen um Elemente der Menge $FORM_{SIG_{OS}}$ aller ontologischen Formeln. Die integritätsbewahrenden Ausdrücke, die mit Hilfe der Typisierungsfunktionen getätigt werden, sind hingegen stets *metasprachlicher* Natur. Sie nehmen nämlich in ihren Argumenten jeweils objektsprachliche Konstrukte auf.

zulässigkeit der Formelmenge FM_1 möglicherweise nur mit der Widersprüchlichkeit der Menge $INT_{SIG_{OS}}$ zusammenhängen könnte. Würden auch widersprüchliche Mengen von Integritätsregeln zugelassen werden, wäre jede Formel $F \in FORM_{SIG_{OS}}$ und jede Formelmenge $FM_1 \subseteq FORM_{SIG_{OS}}$ bezüglich einer Menge $FM_2 \subseteq INT_{SIG_{OS}}$ von Integritätsregeln unzulässig.

Die Widerspruchsfreiheit einer Ontologie $SPEZ_{OS}$ wurde in der Definition durch die Bedingung $(INF_{SIG_{OS}} \cup INT_{SIG_{OS}}) \notin KODM_{SIG_{OS}}$ berücksichtigt. Die Vereinigung der Menge $INF_{SIG_{OS}}$ aller Inferenzregeln mit der Menge $INT_{SIG_{OS}}$ aller Integritätsregeln darf demnach nicht widersprüchlich sein. Dies impliziert, dass weder die Menge $INF_{SIG_{OS}}$ noch die Menge $INT_{SIG_{OS}}$ widersprüchlich sein dürfen. Somit dürfen weder in der Menge $INF_{SIG_{OS}}$ noch in der Menge $INT_{SIG_{OS}}$ widersprüchliche Formeln vorkommen. Darüber hinaus dürfen die Inferenz- und Integritätsregeln einer Ontologie $SPEZ_{OS}$ auch nicht gemeinsam widersprüchlich sein.

Die Überprüfung der Zulässigkeiten von Formeln bezüglich Integritätsregeln wird im Folgenden anhand eines Beispiels verdeutlicht. Gegeben sei dafür die Integritätsregel

$$F_1 = \forall x_1,x_2: R_1(x_1,x_2) \rightarrow \neg R_2(x_1,x_2).$$

Überprüft wird zunächst die Zulässigkeit der Formel

$$F_2 = R_1(t_1,t_2).$$

Die Formel F_2 ist bezüglich der Integritätsregel F_1 zulässig. Die Gültigkeit von F_1 in einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ schließt nämlich die Gültigkeit von F_2 in $A_{SIG_{OS}}$ nicht aus. Ebenso ist die Formel

$$F_3 = R_2(t_1,t_2)$$

bezüglich der Integritätsregel F_1 zulässig. Auch hierbei gilt, dass die Gültigkeit der Integritätsregel F_1 in einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ die Gültigkeit von F_2 in $A_{SIG_{OS}}$ nicht ausschließt. Wenn hingegen die Formelmenge

$$FM = \{F_2,F_3\}$$

betrachtet wird, liegt eine Unzulässigkeit bezüglich F_1 vor, da die Gültigkeit der Integritätsregel F_1 in einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ die Gültigkeit der Formelmenge FM in $A_{SIG_{OS}}$ ausschließt. Für die Vereinigung der drei Formeln gilt nämlich

$$(F_1 \cup F_2 \cup F_3) \in KODM_{SIG_{OS}}.$$

Zu beachten ist, dass nicht nur Grundformeln für eine Integritätsregel-spezifische Überprüfung der Zulässigkeit in Frage kommen, wie das von den o.a. Beispielen suggeriert werden könnte. Mit Hilfe von Integritätsregeln können durchaus auch die Zulässigkeiten von Formeln mit Variablen in ihren Argumenten überprüft werden. Insbesondere von Interesse sind solche variablen Formeln, in denen alle Variablen mittels eines Quantors gebunden sind. Eine solche Überprüfung der Zulässigkeit von variablen Formeln wurde auch schon in Definitionskomponente 4.) für die Widerspruchsfreiheit jeder Ontologie $SPEZ_{OS}$ verwendet. Denn alle Inferenzregeln, deren Zulässigkeit bezüglich aller Integritätsregeln vorausgesetzt wurde, sind variable Formeln.

Beispielsweise ist die Formel

$$F_4 = \forall x_1,x_2: R_1(x_1,x_2) \rightarrow R_3(x_2,x_1)$$

bezüglich der o.a. Integritätsregel F_1 zulässig. Bei Gültigkeit der Integritätsregel F_1 in einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ ist nämlich die Gültigkeit der Formel F_4 in $A_{SIG_{OS}}$ nicht ausgeschlossen. Die Formel

$$F_5 = \forall x_1,x_2: R_1(x_1,x_2) \rightarrow R_2(x_1,x_2)$$

ist hingegen bezüglich F_1 unzulässig. Bei Gültigkeit von F_1 in einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ kann nicht zugleich F_5 in $A_{SIG_{OS}}$ gültig sein.

Integritätsregeln werden öfters in Zusammenhang mit Relationssymbolen benötigt, mit denen Aussagen über Individuen zu Datenkonzepten konstruiert werden. Beispielsweise wird die Antisymmetrie des Relationssymbols *Greater_or_Equal*, das extensional durch die Relation

$$\text{greater\_or\_equal} \subseteq OB_{Real} \times OB_{Real}$$
$$\text{mit greater\_or\_equal}=\{(ob_1,ob_2) \mid ob_1 \geq ob_2\}$$

interpretiert wird, durch die Integritätsregel

$$\forall x_1,x_2: \text{Greater\_Or\_Equal}(x_1,x_2) \wedge \text{Greater\_Or\_Equal}(x_2,x_2) \rightarrow \text{Equal}(x_1,x_2)$$

ausgedrückt. Darüber hinaus können Integritätsregeln allerdings auch dazu verwendet werden, Unzulässigkeiten aufgrund domänenspezifischer Umstände kenntlich zu machen. Beispielsweise kann der Umstand, dass Geschäftsführer eines Unternehmens mindestens 3000 € verdienen,[1] wie folgt ausgedrückt werden:

$$\forall x_1,x_2: \text{Geschaeftsfuehrer\_von}(x_1,x_2) \rightarrow \text{Greater\_Or\_Equal}(\text{hat\_Gehalt}(x_1),3000).$$

Wie eingangs angedeutet, werden durch Regeln Anforderungen an SIG_{OS}-Strukturen gestellt, denen die Strukturen genügen müssen, um eine Ontologie $SPEZ_{OS}$ extensional interpretieren zu können. Dabei wird die Einschränkung der Menge $A(SIG_{OS})$ aller in Frage kommenden Strukturen in erster Linie durch Inferenzregeln bewirkt. Integritätsregeln entfalten im Normalfall ihre Funktionalität erst bei der Überprüfung der Zulässigkeit von wissensrepräsentierenden Formeln. Solche Formeln sind kein Bestandteil einer Ontologie $SPEZ_{OS}$. Die Zulässigkeitsüberprüfung mit Hilfe von Integritätsregeln

1) Das Beispiel ist modifiziert aus RAUH/STICKEL (1997), S. 73, übernommen, um die Vergleichbarkeit der Konstruktionsweisen für Integritätsregeln zu erleichtern. In dem Beispiel werden die Zeichenketten *Geschaeftsfuehrer_von* und *Greater_or_equal* als Relationssymbole und die Zeichenkette *hat_Gehalt* als Operationssymbol verwendet.

wird somit zunächst nur auf Inferenzregeln und auf Integritätsregeln selbst bezogen. Das heißt, dass sowohl die Menge $INF_{SIG_{OS}}$ aller Inferenzregeln als auch die Menge $INT_{SIG_{OS}}$ aller Integritätsregeln aus einer Ontologie $SPEZ_{OS}=(SIG_{OS},INF_{SIG_{OS}},INT_{SIG_{OS}})$ gemeinsam mit der Menge $INT_{SIG_{OS}}$ widerspruchsfrei sein sollten. Würden jedoch bei der Einschränkung der Menge $A(SIG_{OS})$ auf SIG_{OS}-Strukturen nur solche Strukturen ausgeschlossen werden, in denen die Inferenzregeln ungültig sind, so könnten möglicherweise in der Restmenge auch Strukturen enthalten sein, in denen auch die Integritätsregeln ungültig sind. Daher wird die Menge aller SIG_{OS}-Strukturen, mit denen eine Ontologie $SPEZ_{OS}$ extensional interpretiert werden kann, unter Berücksichtigung sowohl aller Inferenz- als auch aller Integritätsregeln bestimmt.

Eine SIG_{OS}-Struktur $A_{SIG_{OS}}$ wird genau dann als *Modell* einer Ontologie $SPEZ_{OS}$ bezeichnet, wenn sowohl alle Inferenzregeln aus der Menge $INF_{SIG_{OS}}$ als auch alle Integritätsregeln aus der Menge $INT_{SIG_{OS}}$ in $A_{SIG_{OS}}$ gültig sind. Die Menge aller Modelle einer Ontologie wird über die Funktion MOD_{OS} bestimmt. Mit ihrer Hilfe kann einer Menge von[1)] ontologischen Formeln die Teilmenge von $A(SIG_{OS})$ zugewiesen werden, in der diese Formeln gültig sind:

$$MOD_{OS}(INF_{SIG_{OS}} \cup INT_{SIG_{OS}}) = \{A_{SIG_{OS}} \mid (A_{SIG_{OS}} \models (INF_{SIG_{OS}} \cup INT_{SIG_{OS}}))\}.$$

Wenn sowohl die Menge $INF_{SIG_{OS}}$ als auch die Menge $INT_{SIG_{OS}}$ aus einer Ontologie $SPEZ_{OS}$ leer sind, stimmt $MOD_{OS}(INF_{SIG_{OS}} \cup INT_{SIG_{OS}})$ mit der Menge $A(SIG_{OS})$ überein. Entsprechend der früheren Feststellung gilt nämlich für jede SIG_{OS}-Struktur $A_{SIG_{OS}} \in A(SIG_{OS})$: $A_{SIG_{OS}} \models \varnothing$.

3.1.4.2 Ontologiegestützte Wissensbasen

In einer Ontologie $SPEZ_{OS}$ werden die sprachlichen Ausdrucksmittel zur Konstruktion von Aussagen und regelartige Zusammenhänge spezifiziert, die in allen Modellen der Ontologie zu gelten haben. In einer Ontologie ist allerdings keine Komponente vorgesehen, um über einen bestimmten Realitätsausschnitt formelartige Aussagen machen zu können. Somit klafft eine Lücke zwischen einerseits einer Ontologie und andererseits dem Realitätsausschnitt, über den mit Hilfe der Ausdrucksmittel einer Ontologie $SPEZ_{OS}$ Aussagen gemacht werden sollen. Diese Lücke wird durch eine *ontologiegestützte Wissensbasis* WBS geschlossen.

1) Es wurde bereits früher darauf hingewiesen, dass die Funktion MOD_{OS} – zwecks einfacherer Diktion – in dem Sinne „überladen“ ist, als dass sie in ihrem Argument sowohl Elemente als auch Teilmengen der Menge $FORM_{SIG_{OS}}$ aller Formeln über einer ontologischen Signatur SIG_{OS} aufnehmen kann.

In einer ontologiegestützten Wissensbasis WBS wird eine (widerspruchsfreie)[1] Ontologie $SPEZ_{OS}$ um eine Menge FB_{EXPL} von ontologischen Grundformeln (Fakten) erweitert. Es handelt sich hierbei um Formeln, mit denen ein Wissensträger sein Wissen über einen Realitätsausschnitt ausdrücken kann. Darüber hinaus ist in einer ontologiegestützten Wissensbasis WBS genau eine SIG_{OS}-Struktur $A_{SIG_{OS}}$ enthalten, in der alle Formeln, mit denen der Wissensträger sein Wissen ausdrückt, gültig sein müssen.

Eine ontologiegestützte Wissensbasis WBS ist definiert als:

$$WBS=(SPEZ_{OS},FB_{EXPL},A_{SIG_{OS}}).$$

Die Komponenten einer ontologiegestützten Wissensbasis WBS sind:

(1.) eine *Ontologie* $SPEZ_{OS}=(SIG_{OS},INF_{SIG_{OS}},INT_{SIG_{OS}})$

(2.) eine *explizite*[2] *Faktenbasis* FB_{EXPL}
mit $FB_{EXPL} \subset GF_{SIG_{OS}}$

(3.) $A_{SIG_{OS}} \in A(SIG_{OS})$
mit $A_{SIG_{OS}} \in MOD_{OS}(INF_{SIG_{OS}} \cup INT_{SIG_{OS}})$
und $A_{SIG_{OS}} \models FB_{EXPL}$.

Die Bezeichnung „ontologiegestützte *Wissens*basis“ lässt es sich zunächst offen, welche Art von Wissen in einer ontologiegestützte Wissensbasis WBS erfasst wird. In der Tat verhält es sich so, dass bei der oben angegebenen Definition von ontologiegestützten Wissensbasen zwei unterschiedliche Wissensarten erfasst werden.

Einerseits wird durch die (widerspruchsfreie) Ontologie $SPEZ_{OS}$ in einer ontologiegestützten Wissensbasis WBS eine spezielle Wissensart erfasst. Es handelt sich hierbei allerdings nicht um das Wissen, dass der Modellierer bezüglich eines Realitätsausschnittes hat. Bei dem Wissen, das in einer Ontologie $SPEZ_{OS}$ enthalten ist, handelt es sich vielmehr um eine „höhergelagerte“ Wissensart, die nicht notwendig einen Bezug auf einen konkreten Realitätsauschnitt aufweisen muss. Das Wissen, das bei der Konstruktion von Ontologien benötigt wird, umfasst alle gerechtfertigten Überzeugungen eines Ontologiekonstrukteurs darüber, welche sprachlichen Ausdruckmittel für die Spezifikation von (Objekt-) Wissen über Realitätsausschnitte in einer vorausgesetzten Domäne benö-

1) Auf die Widerspruchsfreiheit einer Ontologie $SPEZ_{OS}$ wurde im letzten Abschnitt eingegangen. Sie ist genau dann erfüllt, wenn die Vereinigung aus Inferenz- und Integritätsregeln aus $SPEZ_{OS}$ gemeinsam nicht widersprüchlich ist. Die Widerspruchsfreiheit der Ontologie $SPEZ_{OS}$, auf deren Grundlage das Wissensbasierte System WBS spezifiziert ist, wird in den folgenden Ausführungen vorausgesetzt.

2) Der Zusatz „explizite“ wird im folgenden Abschnitt – im Rahmen der Erörterung von Inferenz- und Integritätsregeln – weiter ausgeführt. In diesem Kontext werden nämlich auch *implizite Faktenbasen* als die Menge aller Fakten vorgestellt, die mit Hilfe der Elemente von $INF_{SIG_{OS}}$ aus der expliziten Faktenbasis WB_{EXPL} abgeleitet werden können. Explizite und implizite Faktenbasen werden im Anschluss zu *Wissensbasen* zusammengeführt, die sich dadurch auszeichnen, dass sie sowohl alle explizit angegebenen Fakten als auch alle Ableitungen aus den Fakten und den Inferenzregeln umfassen.

tigt werden. Insofern handelt es sich bei dem Wissen aus einer Ontologie um Metawissen über die Spezifikationsmöglichkeiten für Objektwissen.

Andererseits wird durch die Menge FB_{EXPL} jenes Wissen repräsentiert, das ein Akteur über einen bestimmten Realitätsausschnitt hat. In der Menge FB_{EXPL} sind Aussagen enthalten, die mittels Korrespondenzregeln auf einen Realitätsausschnitt übertragen werden können. Sie werden im Folgenden als *explizite Fakten*[1)] bezeichnet. Explizite Fakten sind ontologische Grundformeln, die in der zugrunde geleten SIG_{OS}-Struktur $A_{SIG_{OS}}$ gültig sind. Insofern handelt es sich bei der expliziten Faktenbasis FB_{EXPL} um eine echte Teilmenge der Menge $GF_{SIG_{OS}}$ aller Grundformeln über der ontologischen Signatur SIG_{OS}.

Die Semantik expliziter Fakten erstreckt sich über zwei Aspekte. Hinsichtlich der *denotationalen Semantik*[2)] wird mit Hilfe expliziter Fakten Wissen über einen bestimmten Realitätsausschnitt repräsentiert. Entsprechend dieser Sichtweise wird z.B. eine atomare Formel $R_j(t_1,...,t_n)$ denotational dadurch interpretiert, dass die realen Objekte, die durch Korrespondenzregeln den formalen Objekten zugeordnet sind, die aus der Auswertung der Terme $t_1,...,t_n$ hervorgehen, in dem repräsentierten Realitätsausschnitt in der entsprechenden Beziehung zueinander stehen, die durch das Relationssymbol R_j repräsentiert wird. Da in Ontologien ein gemeinschaftliches Verständnis über die denotationale Semantik objektsprachlicher Konstrukte vorausgesetzt wird, kann eine eindeutige Korrespondenz der expliziten Fakten zu Annahmen über die Beschaffenheit der Realität unterstellt werden.

Der zweite Aspekt der Semantik expliziter Fakten erstreckt sich über die *formale Semantik*. Hinsichtlich dieses Aspekts gilt für alle expliziten Fakten, dass sie in der SIG_{OS}-Struktur $A_{SIG_{OS}}$ gültig sein müssen, die Bestandteil der ontologiegestützten Wissensbasis WBS ist. Bei der SIG_{OS}-Struktur $A_{SIG_{OS}}$ hat es sich stets um ein Modell aller expliziten Formeln, darunter auch die Inferenz- und Integritätsregeln, aus der zugrunde gelegten Ontologie $SPEZ_{OS}$ zu handeln. Da sowohl die Menge $INT_{SIG_{OS}}$ aller Integritätsregeln als auch die Menge $INF_{SIG_{OS}}$ in der SIG_{OS}-Struktur $A_{SIG_{OS}}$ gültig sein müssen, ist gewährleistet, dass die expliziten Fakten aus FB_{EXPL} zulässig bezüglich aller Integritätsregeln sind.

Aus der Menge FB_{EXPL} aller expliziten Fakten können tendenziell mit Hilfe der Inferenzregeln aus $INF_{SIG_{OS}}$ Fakten abgeleitet werden, die bei Gültigkeit von FB_{EXPL} und $INF_{SIG_{OS}}$ auch gültig sein müssen. Die *Faktenbasis*

$$FB = TM(FB_{EXPL} \cup INF_{SIG_{OS}})$$

---

1) Fakten werden im Singular als „Faktum“ bezeichnet. Darüber hinaus werden die Bezeichner „Fakten“ und „faktische Formeln“ bzw. „Faktum“ und „faktische Formel“ synonym verwendet.

2) Die denotationale Semantik eines sprachlichen Konstrukts wurde in Abschnitt 2.1.3.1 als das außersprachliche realweltliche Konstrukt definiert, durch das es interpretiert wird.

umfasst alle ontologischen Formeln, die bei Gültigkeit von FB_{EXPL} und $INF_{SIG_{OS}}$ auch gültig sein müssen. Entsprechend der Vereinbarung aus dem letzten Abschnitt umfasst die Menge $TM(FB_{EXPL} \cup INF_{SIG_{OS}})$ alle Theoreme der Formelmenge $(FB_{EXPL} \cup INF_{SIG_{OS}})$. Das heißt, dass in einer Faktenbasis FB einerseits alle Formeln aus der expliziten Faktenbasis FB_{EXPL} und andererseits auch alle Inferenzregeln aus $INF_{SIG_{OS}}$ enthalten sind. Darüber hinaus können in der Faktenbasis FB auch solche Formeln enthalten sein, die weder in der expliziten Faktenbasis FB_{EXPL} noch in der Menge $INF_{SIG_{OS}}$ aller Inferenzregeln enthalten sind. Formeln dieser Art werden von der *impliziten* Faktenbasis

$$FB_{IMPL} = FB \setminus (FB_{EXPL} \cup INF_{SIG_{OS}})$$

umfasst. In der Menge FB_{IMPL} aller *impliziten Fakten* sind alle Formeln enthalten, die mit Hilfe der Inferenzregeln aus $INF_{SIG_{OS}}$ aus der expliziten Faktenbasis FB_{EXPL} abgeleitet werden können und weder in FB_{EXPL} noch in $INF_{SIG_{OS}}$ enthalten sind. Es handelt sich hierbei um die Teilmenge der Faktenbasis FB, deren Elemente weder explizite Fakten noch Inferenzregeln sind.

3.1.4.3 Erweiterungen von Ontologien

3.1.4.3.1 Substitution ontologischer Ausdrücke

Für die Entwicklung von Ontologe-Netzen ist eine Erweiterung des bislang vorgestellten formalen Apparats um vier Aspekte notwendig. Hierdurch bleibt der formale Kern von Ontologien, der bislang vorgestellt wurde, unberührt.

Der erste Aspekt erstreckt sich auf die *Substitution* oder *Ersetzung* ontologischer Ausdrücke. Mit Hilfe der Substitution ontologischer Ausdrücke wird die Formulierung von Vor- und Nachbedingungen für das Eintreten von Ereignissen vereinfacht, denen eine zustandsändernde Grundoperation im Modell entspricht.

Aufgrund der Vorgabe der vorliegenden Arbeit, die operationale Semantik mit Hilfe höherer Petri-Netze umzusetzen, wird zudem bei der zweiten Erweiterung eine Ausdrucksvariante vorgestellt, wodurch die Stellen der Petri-Netze am komfortabelsten markiert werden können. Es handelt sich hierbei um *ontologische Termtupel*.

Der dritte und der vierte Aspekt der Erweiterungen lassen sich schließlich als Verfahren der Differenzierung von Relationssymbolen charakterisieren. Bei der ersten Differenzierung werden alle Relationssymbole aussortiert, auf deren Extensionen keine Grundoperationen einwirken können sollen. Relationssymbole dieser Art werden als *statische Relationssymbole* vorgestellt. Sie unterscheiden sich von *dynamischen Relationssymbolen*, deren Extensionen durch Grundoperationen grundsätzlich variiert werden können. Die zweite Differenzierung betrifft schließlich die Unterscheidung zwischen *positiven* und *negativen* Relationssymbolen. Sie ist notwendig, um auch solche Grundformeln aus einer ontologiegestützten Wissensbasis WBS in einem Ontologie-Netz rekonstruieren zu können, die aus der Negation einer atomaren Grundformel hervorgehen.

Die *Substitution* ontologischer Ausdrücke ist ein Verfahren zu deren Transformation, das einerseits aufgrund seiner Bedeutung für das integrative Modellierungskonzept von Interesse ist und andererseits die Anschlussfähigkeit von Ontologien an bewährte Prinzipien der *logischen Programmierung* gewährleistet. Die Substitution wird beispielsweise im Rahmen der *Resolution* dazu verwendet, die Erfüllbarkeit prädikatenlogischer Formeln für maschinelle Theorembeweise zu überprüfen.1) Die Substitution ontologischer Ausdrücke wird durchgeführt, indem zunächst eine Zuordnung vorgenommen wird, durch welche ontologischen Terme jeweils Variablen ersetzt werden sollen. Dieses Verfahren wird im Anschluss so erweitert, dass Variablen aus einem ontologischen Ausdruck $E \in EXPR_{SIG_{OS}}$ durch Terme entsprechend der Zuordnung ersetzt werden. Die Substitution von Variablen durch ontologische Terme erfolgt hierbei durch eine Familie

$$\theta=(\theta_k)_{k \in K}$$

konzeptspezifischer Variablensubstitutionsfunktionen. Eine Familie θ konzeptspezifischer Variablensubstitutionsfunktionen wird verkürzt auch als *Variablensubstitution* bezeichnet. Die Mitglieder

$$\theta_k\text{: } VAR_k \rightarrow TERM_k \text{ für alle } k \in K$$

einer Variablensubstitution θ ordnen allen konzeptspezifischen Variablen $x \in VAR_k$ einen ontologischen Term $\theta_k(x)=t$ mit $t \in TERM_k$ zu.2) Dabei kann eine konzeptspezifische Variable $x \in VAR_k$ sowohl durch einen einfachen als auch durch einen zusammengesetzten ontologischen Term ersetzt werden. Bei der erstgenannten Alternative kann die konzeptspezifische Variable x_1 wiederum durch eine konzeptspezifische Variable $x_2 \in VAR_k$ ersetzt werden. Die Substitution einer Variablen durch eine Variable umfasst auch den Fall, bei dem sich substituierte und substituierende Variable entsprechen.3)

Die Menge 2^θ umfasst alle Variablensubstitutionen.4) Jedes Element $\theta \in 2^\theta$ ist somit eine Familie von konzeptspezifischen Variablensubstitutionsfunktionen θ_k, mittels derer jeder konzeptspezifischen Variablen $x \in VAR_k$ ein Term $t \in TERM_k$ zugeordnet werden kann.

Durch eine Ausdruckssubstitutionsfunktion5)

---

1) Vgl. BEIERLE/KERN-ISBERNER (2000), S. 61 ff.; EBBINGHAUS ET AL. (1992), S. 264 ff.; GALLIER (1986), S. 376 ff.; SCHMID/KINDSMÜLLER (1996), S. 77 ff.; SCHÖNING (1992), S. 88 ff.

2) Vgl. BIBEL (1992), S. 79 f.; EBBINGHAUS ET AL. (1992), S. 63 ff.; SCHÖNING (1992), S. 63 f.

3) Die Substitution von Variablen durch Variablen ist für das hier vorgestellte Modellierungskonzept nicht weiter von Interesse.

4) Vgl. TACKEN (2001), S. 36.

5) Zu beachten ist, dass jede Ausdruckssubstitutionsfunktion $[\theta]$ spezifisch zu einer Familie θ von konzeptspezifischen Variablensubstitutionsfunktionen definiert ist. Die Ausdruckssubstitutionsfunktionen werden im Folgenden auch verkürzt als *Ausdruckssubstitutionen* bezeichnet. Es wird lediglich dann eine Differenzierung vorgenommen, wenn eine Verwechslung mit der Anwendung einer Ausdruckssubstitutionsfunktion auf einen Ausdruck erwartet werden kann.

$$[\theta]: ALPH_{OS}^* \rightarrow ALPH_{OS}^*$$

werden Zeichenfolgen über einem ontologischen Alphabet $ALPH_{OS}$ durch andere Zeichenfolgen über $ALPH_{OS}$ ersetzt. Sie ist spezifisch für eine Familie θ von konzeptspezifischen Variablensubstitutionen definiert.

Als Zeichenfolgen über einem ontologischen Alphabet $ALPH_{OS}$ kommen in erster Linie ontologische Terme und Formeln in Betracht. Dabei werden ontologische Terme durch wiederum ontologische Terme ersetzt. Der ontologische Term $t_2=(t_1)[\theta]$,[1] der aus der konzeptspezifischen Substitution aller Variablen, die in einem ontologischen Term t_1 vorkommen, hervorgeht, ist hierbei dem gleichen Konzept k zugeordnet wie t_1. Darüber hinaus werden ontologische Formeln durch ontologische Formeln ersetzt. Schließlich kann eine Ausdruckssubstitution $[\theta]$ auch dazu verwendet werden, *ontologische Termtupel* durch andere ontologische Termtupel zu ersetzen.[2] Auf diesen Aspekt der Variablensubstitution wird im Rahmen der Analyse ontologischer Termtupel ausführlicher eingegangen.[3]

Die Ausdruckssubstitution $(t)[\theta]$ eines ontologischen Terms $t \in TERM_{SIG_{OS}}$ ist wie folgt definiert:[4]

(1.)	$(x)[\theta]$	$=\theta_k(x)$	für alle $x \in VAR_k$,
(2.)	$(O_i)[\theta]$	$=O_i$	für jedes Konstantensymbol $O_i \in KON_{SIG_{OS}}$ und
(3.)	$(O_i(t_1,...,t_n))[\theta]$	$=O_i(t_1[\theta],...,t_n[\theta])$	für jeden zusammengesetzten ontologischen Term $O_i(t_1,...,t_n) \in TERM_{SIG_{OS}}$.

Die Ausdruckssubstitution $[\theta]$ ordnet einem ontologischen Term $x \in TERM_{SIG_{OS}}$ mit $x \in VAR_k$ den ontologischen Term $t \in TERM_{SIG_{OS}}$ mit $t=(x)[\theta]$ zu, der ihm durch das entsprechende Mitglied θ_k der Familie θ von konzeptspezifischen Variablensubstitutionsfunktionen zugeordnet ist. Die Substitution $(O_i)[\theta]$ eines Konstantensymbols O_i ist hin-

1) Die Bilder der Ausdruckssubstitutionsfunktion werden der gängigen Darstellung folgend in Post-fix-Notation angegeben. Die Anwendung einer Ausdruckssubstitutionsfunktion $[\theta]$ auf eine Zeichenfolge $z \in ALPH_{OS}^*$ wird demnach in der Form $z[\theta]$ angegeben.

2) Ontologische Termtupel werden in Abschnitt 3.1.4.3.2 als Ausdrücke über einer ontologischen Signatur SIG_{OS} vorgestellt. Auf die Substitution ontologischer Termtupel wird in diesem Kontext näher eingegangen. Da eine Ausdruckssubstitutionsfunktion $[\theta]$ auch ontologische Termtupel in ihrem Argument aufnehmen muss, wurde sie nicht als $[\theta]: EXPR_{SIG_{OS}} \rightarrow EXPR_{SIG_{OS}}$ deklariert. Ontologische Termtupel werden nämlich nicht als Elemente der Menge $EXPR_{SIG_{OS}}$ aller ontologischen Ausdrücke eingeführt. Dass dennoch bei einer Ausdruckssubstitution $[\theta]$ ontologische Terme und Formeln auch nur durch ontologische Terme bzw. Formeln substituiert werden, wird bei der Definition der Bilder von $[\theta]$ weiter unten gewährleistet.

3) Vgl. Abschnitt 3.1.4.3.2

4) Vgl. EHRIG ET AL. (1999), S. 356; GALLIER (1986), S. 155; KANEIWA (2001), S. 36 f.; KANEIWA (2004), S. 173.

gegen stets das Konstantensymbol selbst. Zusammengesetzte ontologische Terme werden entsprechend dem Rekursionsschema der Substitutionsregel (3.) ersetzt.

Beispielsweise lautet die Substitution $O_i(x_1,x_2,x_3)[\theta]$ eines zusammengsetzten ontologischen Terms $O_i(x_1,x_2,x_3)$ mit $O_i \in OPS$, $typ_{OPS_{OS}}(O_i)=(k_1k_2k_3,k_4)$ und $x_i \in VAR_{k_i}$ für $1 \leq i \leq 3$ bei einer Ausdruckssubstitution $[\theta]$ mit $[\theta]=(\theta_{k_1},\theta_{k_2},\theta_{k_3})$ und $\theta_{k_i}(x_i)=t_i$ mit $t_i \in TERM_{k_i}$: $O_i(t_1,t_2,t_3)$. Jede Variable x_i aus dem zusammengesetzten ontologischen Term $O_i(x_1,x_2,x_3)$ wird bei der Ausdruckssubstitution $[\theta]$ durch den Term t_i ersetzt, der entsprechend dem jeweiligen Mitglied θ_i der Familie θ der Variable x_i zugeordnet ist.

Bei der Substitution ontologischer Formeln werden lediglich alle *freien* Variablen durch ontologische Terme ersetzt. Variablen, die in einer ontologischen Formel $F \in FORM_{SIG_{OS}}$ gebunden vorkommen, werden hingegen bei der Substitution $[\theta](F)$ „übersprungen". Um dies zu gewährleisten, wird eine *bedingte* Substitutionsfunktion θ_k^x verwendet. Für die bedingte konzeptspezifische Substitutionsfunktion θ_k^x gilt:

$$\theta_k^x(y) = \begin{cases} x & \text{wenn } y = x \\ \theta_k(y) & \text{wenn } y \neq x. \end{cases}$$

Die bedingte konzeptspezifische Substitutionsfunktion θ_k^x ordnet allen Variablen $y \in VAR_k$ einen Wert entsprechend der konzeptspezifischen Substitutionsfunktion θ_k zu, wenn sie nicht mit der ausgezeichneten Variable $x \in VAR_k$ übereinstimmen. Wenn die Variable y im Argument der bedingten Substitutionsfunktion θ_k^x x ist, wird sie „übersprungen". Gemeinsam mit den restlichen konzeptspezifischen Substitutionsfunktionen wird die bedingte Substitutionsfunktion θ_k^x von der Familie θ^x umfasst. Entsprechend wird durch die Mitlgieder der Familie θ^x jeder ontologische Ausdruck durch einen solchen ontologischen Ausdruck substituiert, bei dem frei vorkommende Variablen „übersprungen" wurden.

Die Ausdruckssubstitution $(F)[\theta]$ einer ontologischen Formel $F \in FORM_{SIG_{OS}}$ ist wie folgt definiert:[1)]

(1.)	$w[\theta]$	$=w$,	
(2.)	$f[\theta]$	$=f$,	
(3.)	$(R_j(t_1,\ldots,t_n))[\theta]$	$=R_j(t_1[\theta],\ldots,t_n[\theta])$,	
(4.)	$(\neg F)[\theta]$	$=\neg(F[\theta])$,	
(5.)	$(F_1 \bullet F_2)[\theta]$	$=(F_1)[\theta] \bullet (F_2)[\theta]$	für $\bullet \in \{\wedge,\vee,\underline{\vee},\rightarrow,\leftrightarrow\}$ und
(6.)	$\bullet x:(F)[\theta]$	$=\forall x((F)[\theta^x])$	für $\bullet \in \{\forall,\exists,\underline{\exists}\}$.

Für die tautologische und kontradiktorische Formel w bzw. f sind die Ausdruckssubstitutionen mit sich selbst identisch. Einfache ontologische Formeln werden substituiert,

1) Vgl. EHRIG ET AL. (1999), S. 356; GALLIER (1986), S. 155; KANEIWA (2001), S. 36 f.; KANEIWA (2004), S. 173.

indem alle ontologischen Terme in ihren Argumenten substituiert werden. Für zusammengesetzte ontologische Formeln gelten die Rekursionsschemata der Regeln (4.) bis (6.). Bei quantifizierten Formeln wird dabei, entsprechend dem Rekursionsschema der Regel (6.) die quantifizierte Variable bei der Substitution „übersprungen".

Beispielsweise lautet die Substitution $R_j(x_1,x_2,x_3)[\theta]$ einer ontologischen Formel $R_j(x_1,x_2,x_3)$ mit $R_j \in RS$, $typ_{RS_{OS}}(R_j)=(k_1k_2k_3)$ und $x_i \in VAR_{k_i}$ für $1 \leq i \leq 3$ bei einer Ausdruckssubstitution $[\theta]$ mit $\theta=(\theta_{k_1},\theta_{k_2},\theta_{k_3})$ und $\theta_{k_i}(x_i)=t_i$ mit $t_i \in TERM_{k_i}$: $R_j(t_1,t_2,t_3)$. Jede Variable x_i aus der ontologischen Formel $R_j(x_1,x_2,x_3)$ wird bei der Ausdruckssubstitution $[\theta]$ durch den Term t_i ersetzt, der entsprechend dem jeweiligen Mitglied θ_{k_i} der Variable x_i zugeordnet ist.

Wenn bei einer Ausdruckssubstitution $[\theta]$ eine solche Variablensubstitution θ verwendet wird, die allen frei vorkommenden konzeptspezifischen Variablen $x \in VAR_k$ einen Grundterm $\theta_k(x)=t$ mit $t \in GT_k$ zuordnet, wird $[\theta]$ als *Grundsubstitution* bezeichnet.[1] Bei einer Grundsubstitution $[\theta]$ gelten für alle ontologischen Terme $var_{T_{OS}}((t)[\theta])=\varnothing$ und $fvar_{OS}((F)[\theta])=\varnothing$ für alle ontologischen Formeln. Werden durch eine Ausdruckssubstitution $[\theta]$ zwei ontologische Ausdrücke $E_1,E_2 \in EXPR_{SIG_{OS}}$ so substituiert, dass

$$(E_1)[\theta] = (E_2)[\theta]$$

gilt, wird $[\theta]$ als „Unifikator" von E_1 und E_2 bezeichnet.[2] Wenn eine Ausdruckssubstitution $[\theta]$ existiert, die ein Unifikator zweier ontologischer Ausdrücke $E_1,E_2 \in EXPR_{SIG_{OS}}$ ist, dann wird das Tupel (E_1,E_2) als *unifizierbar* bezeichnet.

3.1.4.3.2 Ontologische Termtupel

Ontologie-Netze werden im weiteren Verlauf als eine Erweiterung von Prädikat/Transition-Netzen (Pr/T-Netze) vorgestellt. Die Klasse der Pr/T-Netze basiert in ihrer ursprünglichen Form auf der konventionellen Prädikatenlogik.[3] Komponenten von Pr/T-Netzen werden in der Regel mit Ausdrücken über einer konventionellen Signatur SIG_{KS} beschriftet. Bei den Ausdrücken, die für die Beschriftung von Pr/T-Netzen verwendet werden, handelt es sich um Termtupel, die die Form $<t_1,...,t_n>$ aufweisen.[4] Sie werden in der Regel über dem n-fachen kartesischen Produkt $\Pi_{n \in \mathbb{N}} TERM_{SIG_{KS}}$ der Menge $TERM_{SIG_{KS}}$ aller Terme über einer konventionellen Signatur SIG_{KS} konstruiert.

Entsprechend dem Vorgehen bei Pr/T-Netzen werden für die Konstruktion von Ontologie-Netzen *ontologische Termtupel*[5] benötigt. Die Menge $TT_{SIG_{OS}}$ umfasst alle Termtu-

---

1) Vgl. DAHR (1994), S. 8; EHRIG ET AL. (1999), S. 355; VAN HOREBEEK/LEWI (1989), S. 28.

2) Vgl. BEIERLE/KERN-ISBERNER (2000), S. 61; BIBEL (1992), S. 89.

3) Vgl. GENRICH (1987), S. 208 ff.

4) Vgl. DAHR (1994), S. 8; KORCZYNSKI ET AL (1990), S. 37.

5) Der Zusatz „ontologische" wird im Folgenden ausgelassen, wenn aus dem Argumentationskontext hervorgeht, dass es sich um Termtupel über einer ontologischen Signatur SIG_{OS} handelt.

pel, die über einer ontologischen Signatur SIG_{OS} konstruiert werden können, und ist wie folgt definiert:[1)]

$$TT_{SIG_{OS}}=\{ <t_1,...,t_n> \mid t_x \in TERM_{SIG_{OS}} \text{ für alle } x=1,...,n \text{ mit } n\in\mathbb{N}_+\}.$$

Die Elemente der Menge $TT_{SIG_{OS}}$ aller ontologischen Termtupel können hinsichtlich der Konzeptzuordnung der bei ihrer Konstruktion verwendeten ontologischen Terme differenziert werden. Jede konzeptfolgenspezifische Menge TT_w mit der Konzeptfolge $w\in K^*$ und $w=(k_1...k_n)$ umfasst alle ontologischen Termtupel $<t_1,...,t_n>$, für die gilt:

$$TT_w=\{ <t_1,...,t_n> \mid w=(k_1...k_n) \wedge t_x\in TERM_{k_x} \text{ für alle } x=1,...,n \text{ mit } n\in\mathbb{N}_+\}.$$

Die Vereinigung aller konzeptfolgenspezifischen Mengen von Termtupeln entspricht stets $TT_{SIG_{OS}}$:

$$TT_{SIG_{OS}} = \cup_{w\in K^*} TT_w.$$

In der Menge $TT_{SIG_{OS}}$ sind alle Termtupel enthalten, die über einer ontologischen Signatur SIG_{OS} konstruiert werden können. Es wird dabei nicht unterschieden, von welcher Art die Terme sind, die in den Termtupeln vorkommen. Die Mengenfamilie

$$TTF_{SIG_{OS}} = (TT_w)_{w\in K^*}$$

umfasst hingegen alle *konzeptfolgen*spezifischen Mengen von Termtupeln, die über einer ontologischen Signatur SIG_{OS} konstruiert werden können. Der x-te ontologische Term t_x mit $x=1,...,n$ und $n\in\mathbb{N}_+$ in einem ontologischen Termtupel $<t_1,...,t_n>\in TT_w$ ist auch stets ein Element der entsprechenden konzeptspezifischen Termmenge $TERM_{k_x}$. Diese konzeptfolgenspezifische Differenzierung ontologischer Termtupel ist ein Merkmal, das ontologische Termtupel von konventionellen Termtupeln abgrenzt. Für Termtupel, die über einer konventionellen Signatur SIG_{KS} konstruiert werden, ist eine solche typgerechte Konstruktion nur bedingt vorgesehen. Konventionelle Termtupel werden lediglich bezüglich der *Anzahl* der Terme, die in einem Termtupel vorhanden sein müssen, differenziert.[2)] Da Terme, die über einer konventionellen Signatur SIG_{KS} konstruierbar sind, grundsätzlich nicht typspezifisch unterschieden werden können, können auch entsprechend über SIG_{KS} keine konzeptfolgenspezifischen Termtupel konstruiert werden.

Der Unterschied zwischen ontologischen und konventionellen Termtupeln kann durch einen Übergang zu einer sortierten Signatur SIG_{SS} überbrückt werden. Dies wird in der Regel bei der Spezifikation von solchen Pr/T-Netzen durchgeführt, die mit Ausdrücken über einer sortierten Signatur SIG_{SS} annotiert werden.[3)] Sortierte Termtupel werden – analog zu ontologischen Termtupeln – über dem kartesischen Produkt sortenspezifischer Termmengen definiert. Entsprechend kann die Menge aller sortierten Termtupel auch hinsichtlich der Typisierung der verwendeten Terme differenziert werden.

1) Vgl. TACKEN (2001), S. 32.

2) Vgl. DAHR (1994), S. 7; JUOPPERI (1995), S. 25 f.; PHILIPPI (1999), S. 163 f.

3) Vgl. KORCZYNSKI ET AL. (1990), S. 37 f.; TACKEN (2001), S. 32; KLEINJOHANN (1993), S. 54 f.

Ein wesentlicher Unterschied von ontologischen gegenüber sortierten Termtupeln liegt in den Strukturierungsrelationen $\sqsubseteq$, $\triangleq$ und $\curlyvee$ begründet. Bei der Definition konzeptspezifischer Termmengen über einer ontologischen Signatur SIG_{OS} werden – neben den Konstruktionsregeln, die auch für sortenspezifische Termmengen Gültigkeit haben – auch die metasprachlichen Strukturierungsrelationen $\sqsubseteq$, $\triangleq$ und $\curlyvee$ berücksichtigt.[1] Mit Hilfe der metasprachlichen Strukturierungsrelationen werden Beziehungen zwischen Konzepten spezifiziert, aufgrund derer die entsprechenden konzeptspezifischen Termmengen in Teilmengen-, Gleichheits- bzw. Disjunktheitsbeziehungen zueinander stehen müssen. Dieser Aspekt der Konstruktion ontologischer Terme wirkt sich wesentlich auf die Konstruktion ontologischer Termtupel aus. Eine konzeptfolgenspezifische Termtupelmenge TT_w mit $w=(k_1...k_n)$ umfasst nämlich alle ontologischen Termtupel $<t_1,...,t_n>$, für die $t_x \in TERM_{k_x}$ für alle $x=1,...,n$ mit $n \in \mathbb{N}_+$ gilt. Hierzu gehören neben allen ontologischen Termtupeln, die aufgrund der originären Konzeptzuordnung ihrer Terme zu TT_w gezählt werden, auch alle Termtupel, deren Terme aufgrund einer metasprachlichen Strukturierungsbeziehung $\sqsubseteq$ oder $\triangleq$ zwischen den entsprechenden Konzepten derivativ zu TT_w gezählt werden.

Ein ontologisches Termtupel $<t_1,...,t_n>$ kann demnach zu einer konzeptfolgenspezifischen Menge TT_w gezählt werden, wenn die ontologischen Terme $t_x \in TERM_{SIG_{OS}}$ zu den konzeptspezifischen Termmengen $TERM_{k_x}$ aufgrund ihrer originären Zuordnung oder ihrer derivativen Zuordnung wegen der Subkonzeptrelation $\sqsubseteq$ oder der Äquivalenzrelation $\triangleq$ gezählt werden. Somit kann ein ontologisches Termtupel $<t_1,...,t_n>$ in zwei konzeptfolgenspezifischen Mengen TT_{w_1} und TT_{w_2} mit $w_1=(k_{1_1}...k_{1_n})$, $w_2=(k_{2_1}...k_{2_n})$ und $k_{1_x} \neq k_{2_x}$ für mindestens ein $x=1,...,n$ mit $n \in \mathbb{N}_+$ enthalten sein. Dies ist z.B. dann der Fall, wenn die Konzeptfolge $w_1 \in K^*$ der Konzeptfolge $w_2 \in K^*$ *untergeordnet* ist. Dabei wird die Ordnungsbeziehung zwischen Konzeptfolgen durch die Relation

$$\boxed{\begin{array}{l} \qquad\qquad\qquad \preceq \;\subseteq\; (K^* \times K^*) \\ \text{mit } \preceq \;=\; \left\{\left(\left(k_{a_1}...k_{a_n}\right),\left(k_{b_1}...k_{b_n}\right)\right) \mid \left(k_{a_i} \sqsubseteq k_{b_i}\right) \vee \left(k_{a_i} \triangleq k_{b_i}\right) \vee \left(k_{a_i} = k_{b_i}\right)\right\} \end{array}}$$

ausgedrückt. Das Tupel $(\preceq,K^*)$ erfüllt die Eigenschaften der

(1.) Reflexivität: $\forall w \in K^*: w \preceq w$ und

(2.) Transitivität $\forall w_1,w_2,w_3: ((w_1 \preceq w_2) \wedge (w_2 \preceq w_3)) \rightarrow (w_1 \preceq w_3)$.[2]

---

1) Vgl. Abschnitt 3.1.2.2.1.

2) Die Eigenschaften der Reflexivität und Transitivität entsprechen dem „kleinsten gemeinsamen Nenner“, die einerseits die Strukturierungsrelationen $\sqsubseteq$ und $\triangleq$ sowie andererseits die Gleichheitsrelation = erfüllen. Sowohl die Reflexivität als auch Transitivität gelten nämlich für alle drei Relationen, deren adjunktive Verknüpfung in der Definition von $\preceq$ vorkommt. Hinsichtlich der Symmetrie von $\preceq$ können hingegen keine Aussagen gemacht werden, da die Subkonzeptrelation $\sqsubseteq$ im Gegensatz zu $\triangleq$ antisymmetrisch ist.

Die konzeptfolgenspezifische Menge TT_{w_1} von ontologischen Termtupeln ist aufgrund der Semantik der metasprachlichen Strukturierungsrelationen eine Teilmenge der zweiten konzeptfolgenspezifischen Menge TT_{w_2} von ontologischen Termtupeln, wenn die Konzeptfolge w_1 der Konzeptfolge w_2 untergeordnet ist:

$$\forall w_1,w_2 \in K^*: (w_1 \preceq w_2) \rightarrow TT_{w_1} \subseteq TT_{w_2}.$$

Dies wird anhand von drei Beispielen diskutiert:[1)] Gegeben sei eine konzeptfolgenspezifische Menge TT_w mit $w=(k_1,...,k_n)$. Ein ontologisches Termtupel $tt_a=<t_{a_1},...,t_{a_n}>$, bei dem für alle verwendeten ontologischen Terme $t_{a_x} \in KON_{k_x}$ gilt, ist ein Element von TT_w, da

$$KON_{k_x} \subseteq TERM_{k_x} \text{ für } x=1,...,n \text{ mit } n \in \mathbb{N}_+$$

gilt. In diesem Fall gehört das Termtupel $<t_{a_1},...,t_{a_n}>$ zu TT_w, da alle ontologischen Terme t_{a_x} *originär* den entsprechenden konzeptspezifischen Termmengen $TERM_{k_x}$ zugeordnet sind. Die originäre Zuordnung eines ontologischen Terms t_{a_x} ist durch $typ_{T_{OS}}(t_{a_x})$ angegeben. Somit gilt $<t_{a_1},...,t_{a_n}> \in TT_{k_1...k_n}$, wenn $typ_{T_{OS}}(t_{a_x})=k_x$ für alle $x=1,...,n$ und $n \in \mathbb{N}_+$ gilt.

Ein zweites Termtupel $tt_b=<t_{b_1},...,t_{b_n}>$ ist ein Element von TT_w mit $w=(k_1...k_n)$, wenn für alle verwendeten ontologischen Terme $t_{b_x} \in KON_{k_{b.x}}$

$$(k_{b_x} \neq k_x) \text{ und } (k_{b_x} \sqsubseteq k_x) \text{ für } x=1,...,n \text{ mit } n \in \mathbb{N}_+$$

gelten. Dann gilt nämlich $KON_{k_{b.x}} \subseteq TERM_{k_x}$ für $x=1,...,n$. In diesem Fall wird das Termtupel $<t_{b_1},...,t_{b_n}>$ zu TT_w gezählt, weil alle ontologischen Terme t_{b_x} originär zu solchen konzeptspezifischen Termmengen $TERM_{k_{b.x}}$ gezählt werden, für die

$$TERM_{k_{b.x}} \subseteq TERM_{k_x} \text{ für } x=1,...,n \text{ mit } n \in \mathbb{N}_+$$

gilt. Somit gilt $<t_{b_1},...,t_{b_n}> \in TT_{k_1...k_n}$, wenn zwar einerseits $typ_{T_{OS}}(t_{b_x})=k_{b_x}$ mit $k_{b_x} \neq k_x$ gilt, allerdings andererseits $(k_{b_x} \sqsubseteq k_x)$ gilt.

Ähnlich zum letzten Beispiel wird ein drittes Termtupel $<t_{c_1},...,t_{c_n}>$, bei dem für alle verwendeten ontologischen Terme $t_{c_x} \in KON_{k_x}$

$$(k_{c_x} \neq k_x) \text{ und } (k_{c_x} \triangleq k_x) \text{ für } x=1,...,n \text{ mit } n \in \mathbb{N}_+$$

gelten, zu TT_w gezählt. In diesem Fall wird $<t_{c_1},...,t_{c_n}>$ zu $TT_{k_1...k_n}$ gezählt, weil die Terme $t_{c_x} \in TERM_{k_{c.x}}$ aufgrund der Äquivalenzrelation $\triangleq$ auch zu allen konzeptspezifischen Termmengen $TERM_{k_x}$ gezählt werden.

1) Aus Gründen der Einfachheit werden bei den Beispielen lediglich atomare ontologische Terme verwendet, deren originäre Konzeptzuordnung kenntlich gemacht wird. Die Darstellung lässt sich allerdings auch auf zusammengesetzte ontologische Terme verwenden, die aus der typgerechten Anwendung von Operationssymbolen auf andere ontologische Terme hervorgehen.

Ontologische Termtupel werden in Ontologie-Netzen zu verschiedenden Zwecken benötigt. Für einen Verwendungszweck[1)] werden lediglich solche ontologischen Termtupel zugelassen, die nur mit Hilfe von ontologischen *Grund*termen konstruiert wurden. Das heißt, dass für den Verwendungszweck ontologische Termtupel, in denen auch variable ontologische Terme vorkommen, ausgeschlossen sind. Um solche Termtupel kenntlich machen zu können, wird die Menge $GTT_{SIG_{OS}}$ aller *ontologischen Grundtermtupel* über der ontologischen Signatur SIG_{OS} eingeführt. Es handelt sich hierbei um solche ontologischen Termtupel, zu deren Konstruktion nur ontologische Grundterme verwendet werden:

$$GTT_{SIG_{OS}} \subseteq TT_{SIG_{OS}}$$
$$GTT_{SIG_{OS}}=\{ <t_1,...,t_n> \mid t_x \in GT_{SIG_{OS}} \text{ für alle } x=1,...,n \text{ mit } n \in \mathbb{N}_+\}.$$

Die Menge $GTT_{SIG_{OS}}$ aller ontologischen Grundtermtupel über einer ontologischen Signatur SIG_{OS} kann hinsichtlich der Typen der verwendeten ontologischen Terme differenziert werden:

$$GTT_{SIG_{OS}} = \cup_{w \in K^*} GTT_w$$
$$\text{mit } GTT_w=\{<t_1,...,t_n> \mid w=(k_1,...,k_n) \text{ und } t_x \in GT_{k_x} \text{ für alle } x=1,...,n \text{ und } n \in \mathbb{N}_+\}.$$

Die Mengenfamilie

$$GTTF_{SIG_{OS}} = (GTT_w)_{w \in K^*}$$

umfasst alle konzeptfolgenspezifische Mengen von Grundtermtupeln, die über einer ontologischen Signatur SIG_{OS} konstruiert werden können.

Wie bei jeder Menge, können die Elemente in den Mengen $TT_{SIG_{OS}}$ und $GTT_{SIG_{OS}}$ höchstens einmal vorkommen. Für die Zwecke, bei denen ontologische Termtupel in Ontologie-Netzen benötigt werden, kann allerdings auch das mehrfache Vorkommen desselben ontologischen Termtupels notwendig sein. Für diese Zwecke werden *Multimengen* über den Mengen $TT_{SIG_{OS}}$ bzw. $GT_{SIG_{OS}}$ verwendet. In einer Multimenge

$$mult_1: TT_{SIG_{OS}} \rightarrow \mathbb{N}$$

kommt ein ontologisches Termtupel $tt_1 \in TT_{SIG_{OS}}$ genau $mult_1(tt_1)$ mal vor. Entsprechend kommt in einer Multimenge

$$mult_2: GTT_{SIG_{OS}} \rightarrow \mathbb{N}$$

ein ontologisches Grundtermtupel gtt_2 genau $mult_2(gtt_2)$ mal vor. Die Mengen $MULT(TT_{SIG_{OS}})$ und $MULT(GTT_{SIG_{OS}})$ umfassen jeweils alle Multimengen über der Menge $TT_{SIG_{OS}}$ aller ontologischen Termtupel bzw. über der Menge $GTT_{SIG_{OS}}$ aller ontologischen Grundtermtupel über einer ontologischen Signatur SIG_{OS}.

---

1) Es handelt sich bei dem angesprochenen Verwendungszweck um die Markierung von *Stellen* mit Grundtermtupeln. Sowohl Stellen aus auch ihre Markierung in Ontologie-Netzen werden in Abschnitt 4.2 vorgestellt.

Die Menge der Variablen, die in einer Multimenge $mult \in MULT(TT_{SIG_{OS}})$ vorkommen, wird durch die Funktion

$$var_{TT}: MULT(TT_{SIG_{OS}}) \rightarrow pot(VAR)$$

bestimmt. Dabei gelten:

(1.) $var_{TT}(<t_1,...,t_n>)$
$= \cup_{x=1,...,n} var_{T_{OS}}(t_x)$ und

(2.) $var_{TT}(<t_{1_1},...,t_{1_{l(1)}}> +...+ <t_{X_1},...,t_{X_{l(X)}}>)$
$= var_{TT}(<t_{1_1}...t_{1_{l(1)}}>) \cup...\cup var_{TT}(<t_{X_1}...t_{X_{l(X)}}>)$.

Entsprechend der Substitution ontologischer Terme und Formeln können mit Hilfe einer Ausdruckssubstitution $[\theta]$ mit zugrunde liegender Variablensubstitution θ auch ontologische Termtupel substituiert werden. Denn die Ausdruckssubstitution $[\theta]$ kann alle Zeichenfolgen, die über dem formalsprachlichen Alphabet $ALPH_{OS}$ konstruiert sind, in ihrem Argument aufnehmen. Bezogen auf Multimengen von Termtupeln ist ihre Substitution wie folgt definiert:

(1.) $(<t_1,...,t_n>)[\theta]$
$= <(t_1)[\theta],...,(t_n)[\theta]>$ und

(2.) $(<t_{1_1},...,t_{1_{l(1)}}> +...+ <t_{I_1},...,t_{I_{l(I)}}>)[\theta]$
$= <(t_{1_1})[\theta],...,(t_{1_{l(1)}})[\theta]> +...+ <(t_{I_1})[\theta],...,(t_{I_{l(I)}})[\theta]>$.

Eine Multimenge der Form $mult_1=<t_1,...,t_n>$ von ontologischen Termtupeln wird durch eine Ausdruckssubstitution $[\theta]$ substituiert, indem jeder ontologische Term t_i mit $i=1,...,n$ und $n \in \mathbb{N}_+$ durch $t_i[\theta]$ ersetzt wird. Beispielsweise lautet die Substitution $(<x_1,...,x_n>)[\theta]$ einer Multimenge $<x_1,...,x_n> \in MULT(TT_{SIG_{OS}})$ ontologischer Termtupel bei einer Familie $\theta=(\theta_1,\theta_2,\theta_3)$ von konzeptspezifischen Variablensubstitutionen mit $\theta_i(x_i)=t_i$, $i=1,...,n$ und $n \in \mathbb{N}_+$: $<t_1,...,t_n>$

Eine Multimenge der Form

$$mult_2=<t_{1_1},...,t_{1_{l(1)}}> +...+ <t_{I_1},...,t_{I_{l(I)}}>$$

wird durch eine Ausdruckssubstitution $[\theta]$ substituiert, indem jeder ontologische Term t_{i_j} mit $i=1,...,I$, $I \in \mathbb{N}_+$, $j=1,...,max\{l(1),...,l(I)\}$ durch $(t_{i_j})[\theta]$ ersetzt wird. Beispielsweise lautet die Substitution $(<x_{1_1},...,x_{1_{l(1)}}>+...+<x_{I_1},...,x_{I_{l(I)}}>)[\theta]$ einer Multimenge $<x_{1_1},...,x_{1_{l(1)}}>+...+<x_{I_1},...,x_{I_{l(I)}}> \in MULT(TT_{SIG_{OS}})$ ontologischer Termtupel bei einer Familie $\theta=(\theta_1,\theta_2,\theta_3)$ von konzeptspezifischen Variablensubstitutionen mit $\theta_{i_j}(x_{i_j})=t_{i_j}$, $i=1,...,I$, $I \in \mathbb{N}_+$, $j=1,...,max\{l(1),...,l(I)\}$: $<t_{1_1},...,t_{1_{l(1)}}> +...+ <t_{I_1},...,t_{I_{l(I)}}>$.

Analog zu der Auswertung von ontologischen Termen und ontologischen Formeln wird schließlich ein Verfahren benötigt, um ontologische Termtupel auswerten zu können. Ontologische Termtupel sind bislang lediglich als syntaktische Konstrukte definiert, denen ohne ein Auswertungsverfahren keine formale Bedeutung zugewiesen werden kann. Mit Hilfe eines Auswertungsverfahrens wird ontologischen Termtupeln eine formale Semantik zugewiesen.

Die formale Semantik wird im Folgenden auf Multimengen von ontologischen Termtupeln bezogen, da lediglich solche in Ontologie-Netzen Anwendung finden. Da jede konventionelle Menge A auch eine Multimenge mult(A) mit der Kardinalität mult(a)=1 für jedes Element $a \in A$ ihrer Trägermenge A ist, kann das Auswertungsverfahren auch für konventionelle Mengen ontologischer Termtupel angewendet werden.

Für die Auswertung von ontologischen Termtupeln in einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ wird eine Familie

$$IF_{TT} = (I_w)_{w \in K^*}$$

von *konzeptfolgenspezifischen Termtupelauswertungsfunktionen* herangezogen. Jede konzeptfolgenspezifische Termtupelauswertungsfunktion I_w mit $w \in K^*$, $w=(k_1...k_n)$ und $n \in \mathbb{N}_+$ ordnet einer Multimenge $mult_1 \in MULT(TT_{k_1...k_n})$ über der konzeptfolgenspezifischen Menge TT_w eine Multimenge $mult_2 \in MULT(OB_{k_1} \times ... \times OB_{k_n})$ zu:[1]

$$I_{k_1...k_n}: MULT(TT_{k_1...k_n}) \rightarrow MULT(OB_{k_1} \times ... \times OB_{k_n}).^{2)}$$

Dabei ist die Multimenge $mult_2$, auf die die Multimenge $mult_1$ abgebildet wird, über dem kartesischen Produkt $OB_{k_1} \times ... OB_{k_n}$ der konzeptspezifischen Objektmengen $OB_{k_1},...,OB_{k_n}$ aus der zugrunde gelegten SIG_{OS}-Struktur $A_{SIG_{OS}}$ definiert. Die Bilder jeder konzeptfolgenspezifischen Termtupelauswertungsfunktion $I_{k_1...k_n}$ gehen hierbei aus den Bildern der konzeptspezifischen Termauswertungsfunktionen IT_{k_x} mit x=1,...,n und $n \in \mathbb{N}_+$ hervor, die wiederum auf den extensionalen Interpretationsfunktionen aus IF_{OS} basieren. Sie sind bei einer Familie ITF_{OS} von konzeptspezifischen Termauswertungsfunktionen wie folgt definiert:[3]

(1.) Für $mult_1 = <t_1,...,t_n>$ mit $t_x \in TERM_{k_x}$ für x=1,...,n und $n \in \mathbb{N}_+$:
$I_{k_1...k_n}(mult_1) \in (OB_{k_1} \times ... \times OB_{k_n} \rightarrow \mathbb{N})$ mit
$I_{k_1...k_n}(<t_1,...,t_n>) = (ob_1,...,ob_n)$
mit $ob_x = IT_{k_x}(t_x)$ für alle x=1,...,n,

(2.) für $mult_1 = z<t_1,...,t_n>$ mit $z \in \mathbb{N}_+$, $z \geq 2$ und $t_x \in TERM_{k_x}$ für x=1,...,n und $n \in \mathbb{N}_+$:
$I_{k_1...k_n}(mult_1) \in (OB_{k_1} \times ... \times OB_{k_n} \rightarrow \mathbb{N})$ mit
$I_{k_1...k_n}(z<t_1,...,t_n>) = z(I_{k_1...k_n}(<t_1,...,t_n>))$ und

(3.) Für $mult_1 = z_1<t_{1_1},...t_{1_n}> + ... + z_A<t_{A_1},...,t_{A_n}>$ mit a=1,...,A, $a \in \mathbb{N}_+$ und $t_{a_x} \in TERM_{k_x}$ für x=1,...,n:
$I_{k_1...k_n}(mult_1) \in (OB_{k_1} \times ... \times OB_{k_n} \rightarrow \mathbb{N})$ mit

$$\begin{aligned} I_{k_1...k_n}(mult_1) &= I_{k_1...k_n}(z_1<t_{1_1},...t_{1_n}>) \oplus ... \oplus I_{k_1...k_n}(z_A<t_{A_1},...t_{A_n}>) \\ &= z_1(I_{k_1...k_n}(<t_{1_1},...t_{1_n}>)) \oplus ... \oplus z_A(I_{k_1...k_n}(z_A<t_{A_1},...t_{A_n}>)). \end{aligned}$$

---

1) Multimengen von Termtupeln, in denen Termtupel unterschiedlichen Typs – als einer unterschiedlichen Konzeptfolge – vorkommen, werden in der vorliegenden Arbeit nicht berücksichtigt. Daher wird die Auswertung nur in Bezug auf Multimengen von Termtupeln aufgezeigt, in denen nur Termtupel gleichen Typs vorkommen.

2) Vgl. KORCZYNSKI ET AL. (1990), S. 38.

3) Vgl. TACKEN (2001), S. 38 f.

Eine konzeptfolgenspezifische Termtupelauswertungsfunktion I_w mit $w \in K^*$ ordnet gemäß (1.) einer Multimenge $mult_1$, in der nur ein ontologisches Termtupel $<t_1,...,t_n>$ enthalten ist, das Objekttupel $(ob_1,...,ob_n)$ zu, das aus der Auswertung $IT_{k_x}(t_x)$ jedes x-ten ontologischen Terms t_x zum formalen Objekt ob_x hervorgeht.

Eine Multimenge $mult_1=z<t_1,...,t_n>$, in der das ontologische Termtupel $<t_1,...,t_n>$ z mal vorkommt, wird gemäß (2.) zu einer Multimenge ausgewertet, in der das Objekttupel $(ob_1,...,ob_n)$, das wiederum aus der konzepttreuen Auswertung $IT_{k_x}(t_x)$ jedes x-ten ontologischen Terms t_x zum Individuum ob_x hervorgeht, z mal vorkommt.

Eine Multimenge $mult_1=z_1<t_{1_1},...t_{1_n}>+...+z_A<t_{A_1},...,t_{A_n}>$, in der jedes ontologische Termtupel $<t_{a_1},...,t_{a_n}>$ mit $a=1,...,A$ und $A \in \mathbb{N}_+$ z_a mal vorkommt, wird gemäß (3.) zu einer Multimenge von Objekttupeln ausgewertet, die aus der komponentenweisen Addition jeder Multimenge $I_w(z_a<t_{a_1},...,t_{a_n}>)$ hervorgeht. Bei der komponentenweisen Addition von Multimengen werden die Summen der multimengenspezifischen Multiplizitäten für Elemente angesetzt wird.[1)]

1) Vgl. Abschnitt 2.2.2.

3.1.4.3.3 Differenzierung der Relationssymbole

3.1.4.3.3.1 Statische und dynamische Relationssymbole

Das zentrale Anliegen der vorliegenden Arbeit liegt darin, die ansonsten zeitlich invarianten Interpretationen von Ontologien um *dynamische* Aspekte anzureichern. Die zeitliche Invarianz der Interpretation von Ontologien äußert sich in erster Linie darin, dass für Ontologien keine Auszeichnung des *Zustandes* vorgesehen ist, in dem die Formeln die mit den sprachlichen Ausdrucksmitteln konsturiert werden können, gültig sind. Insofern handelt es sich bei den bislang vorgestellten SIG_{OS}-Strukturen üblicherweise um *statische SIG_{OS}-Strukturen*. Die Statik von SIG_{OS}-Strukturen kommt explizit in der *Monotonieprämisse* zur Geltung, die prädikatenlogischen Ansätzen zugrunde liegt. Ihr zufolge bleibt die Gültigkeit einer Formel(-menge) in einer Struktur auch bei Hinzufügen weiterer Formeln für immer erhalten. Diese Starrheit der prädikatenlogischen Wissensmodellierung steht allerdings im Konflikt mit dem Vorhaben, die Gültigkeit ontologischer Formeln im Zeitverlauf variieren zu können. Ein zentraler Aspekt der Erweiterung von Ontologien um dynamische Aspekte liegt nämlich darin, die Gültigkeit oder Ungültigkeit von wissensrepräsentierenden Formeln variieren zu können.

Grundsätzlich sind Versuche zur Überbrückung der zeitlichen Starrheit der Gültigkeit prädikatenlogischen Wissensrepräsentation mit erheblichen Schwierigkeiten verbunden. Für Wissensrepräsentationen, die auf dem formalen Rahmen der Prädikatenlogik basieren, ist nämlich die Auszeichnung des *Zustands*, in dem die wissensrepräsentierenden Formeln entweder gültig oder ungültig sind, mit einer Verkomplizierung der modelltheoretischen Semantik verbunden. In der modelltheoretischen Semantik – die auch Ontologien zugrunde liegt – wird nämlich die formale Semantik eines Ausdrucks – im Fall von Termen – durch eine Abbildung auf ein Individuum aus einer SIG_{OS}-Struktur und – im Fall von Formeln – durch die Bestimmung ihrer Gültigkeit in einer SIG_{OS}-Struktur angegeben. Dabei wird weder für die Termauswertungsfunktionen noch für die Modellrelation ihre *Zustandsspezifität* angegeben.

Im Fall zeitlich variabler Strukturen zu Ontologien ist jedoch eine Angabe des Zustands, in dem die betrachtete Auswertung gültig ist, notwendig. Beispielsweise kann eine Formel $R_j(t_1,...,t_n)$ in einem ersten Zustand des Systems in einer SIG_{OS}-Struktur $A_{SIG_{OS.1}}$ gültig sein, allerdings in einem zweiten Zustand des Systems in einer SIG_{OS}-Struktur $A_{SIG_{OS.2}}$ ungültig sein. Entsprechend ist eine Variation des Gültigkeitswerts von Formeln, die mit einem Relationssymbol R_j konstruiert sind, nur dann denkbar, wenn auch die Extension von R_j variieren kann.

Um die zeitliche Varianz von einzelnen Relationssymbolextensionen formal erfassen zu können, wird die Menge RS aller Relationssymbole aus dem formalsprachlichen Alphabet $ALPH_{OS}$ in die zueinander disjunkten Mengen RS_{ST} und RS_{DY} unterteilt:

$$RS = RS_{ST} \cup RS_{DY}$$
$$\text{mit } RS_{ST} \cap RS_{DY} = \varnothing.$$

Die Menge RS_{ST} umfasst solche Relationssymbole aus dem formalsprachlichen Alphabet $ALPH_{OS}$, deren Extensionen zeitlich invariant sind. In der Menge RS_{DY} sind hingegen solche Relationssymbole aus dem formalsprachlichen Alphabet $ALPH_{OS}$ enthalten, deren Extensionen zeitabhängig variieren können.

Mit der Differenzierung der Menge RS aller Relationssymbole geht auch eine Differenzierung der ontologischen Signatur SIG_{OS} einher, in der RS enthalten ist. Eine *dynamische ontologische Signatur*[1)]

$$SIG_{DY}=(K,OPS,RS_{DY},typ_{OPS_{OS}},typ_{RS_{DY}},....)$$

geht aus einer ontologischen Signatur $SIG_{OS}=(K,OPS,RS,typ_{OPS_{OS}},typ_{RS_{OS}},...)$ dadurch hervor, dass einerseits die Menge RS_{DY} eine Teilmenge der Menge RS und andererseits die Funktion $typ_{RS_{DY}}$ eine Teilmenge der Funktion $typ_{RS_{OS}}$ ist. Dabei lautet die Funktionsvorschrift für $typ_{RS_{DY}}$:

$$typ_{RS_{DY}}: RS_{DY} \rightarrow K^{+}.$$

Komplementär zu SIG_{DY} geht die *statische ontologische Signatur*[2)]

$$SIG_{ST}=(K,OPS,RS_{ST},typ_{OPS_{OS}},typ_{RS_{ST}},....)$$

aus der ontologischen Signatur SIG_{OS} dadurch hervor, dass einerseits nur Relationssymbole mit zeitlich invarianten Extensionen berücksichtigt werden und andererseits nur die dazu „passenden" Elemente der Typisierungsfunktion $typ_{RS_{ST}}$ aufgenommen werden. Ihre Vorschrift lautet:

$$typ_{RS_{ST}}: RS_{ST} \rightarrow K^{+}.$$

Entsprechend der Differenzierung der ontologischen Signatur SIG_{OS} wird auch die Menge $EXPR_{SIG_{OS}}$ aller Ausdrücke über SIG_{OS} unterteilt. Sie betrifft allerdings nur die Menge $FORM_{SIG_{OS}}$ aller Formeln über der ontologischen Signatur SIG_{OS}, da Relationssymbole bei der Konstruktion von ontologischen Termen nicht berücksichtigt werden und die Differenzierung nur Relationssymbole betrifft. Die Unterteilung der Menge $FORM_{SIG_{OS}}$ aller Formeln über der ontologischen Signatur SIG_{OS} lautet:

$$(FORM_{SIG_{ST}} \cup FORM_{SIG_{DY}}) \subseteq FORM_{SIG_{OS}}$$
$$\text{mit } FORM_{SIG_{DY}} \cap FORM_{SIG_{ST}} = \varnothing.$$

Die Menge $FORM_{SIG_{DY}}$ umfasst solche ontologischen Formeln, bei deren Konstruktion nur dynamische Relationssymbole aus der Menge RS_{DY} aus SIG_{DY} verwendet werden. Die dazu komplementäre Menge $FORM_{SIG_{ST}}$ umfasst alle Formeln über der ontologischen Signatur SIG_{OS}, in denen nur statische Relationssymbole aus RS_{ST} vorkommen.

1) Wenn aus dem Argumentationskontext hervorgeht, dass es sich um eine dynamische ontologische Signatur SIG_{DY} handelt, werden auch die Schreibweisen „ontologische Signatur" oder „dynamische Signatur" zugelassen.

2) Wenn aus dem Argumentationskontext hervorgeht, dass es sich um eine statische ontologische Signatur SIG_{ST} handelt, werden auch die Schreibweisen „ontologische Signatur" oder „statische Signatur" zugelassen.

Ontologische Formeln, die weder in $FORM_{SIG_{DY}}$ noch in $FORM_{SIG_{ST}}$ enthalten sind, enthalten sowohl dynamische als auch konventionelle Relationssymbole. Während die Gültigkeit von Formeln aus $FORM_{SIG_{ST}}$ in einer Struktur zeitlich invariant ist, da für ihre Konstruktion nur solche Relationssymbole verwendet werden, die zeitlich invariante Extensionen aufweisen, kann die Gültigkeit von Formeln aus $FORM_{SIG_{DY}}$ zustandsspezifisch variieren.

Sonderfälle bei der Differenzierung der Menge $FORM_{SIG_{OS}}$ aller ontologischen Formeln über SIG_{OS} sind die tautologische und die kontradiktorische Formel w bzw. f. Für beide Formeln wird angenommen, dass sie zeitlich invariante Gültigkeit haben. Daher werden sowohl w als auch f zu der Menge $FORM_{SIG_{ST}}$ aller statischen ontologischen Formeln gezählt:

$$w,f \in FORM_{SIG_{ST}}.$$

Der Differenzierung der Signatur SIG_{OS} wird eine Differenzierung der SIG_{OS}-Strukturen gegenübergestellt, mit denen die ontologische Signatur SIG_{OS} extensional interpretiert wird. Jede SIG_{OS}-Struktur $A_{SIG_{OS}}$ kann entsprechend der Differenzierung der Relationssymbole aus SIG_{OS} aufgeteilt werden. Der extensionalen Interpretation einer statischen ontologischen Signatur SIG_{ST}, in der nur solche Relationssymbole berücksichtigt werden, die eine zeitlich invariante Extension aufweisen, entspricht nunmehr einer *statischen SIG_{OS}-Struktur*

$$A_{SIG_{ST}}=(OBF_{OS},OPF,RF_{ST},IF_{ST})$$
$$\text{mit } RF_{ST} \subseteq RF$$
$$\text{und } IF_{ST}=(I_K,I_{OPS},I_{RS_{ST}}).$$

Für die extensionale Interpretation einer statischen ontologischen Signatur SIG_{ST} werden die Interpretationsfunktionen $IF_{ST}=(I_K,I_{OPS},I_{RS_{ST}})$ verwendet. Dabei bildet die extensionale Interpretationsfunktion

$$I_{RS_{ST}}: RS_{ST} \rightarrow RF_{ST}$$

auf bijektive Weise jedes statische Relationssymbol $R_j \in RS_{ST}$ auf genau eine statische Relation $r_j \in RF_{ST}$ ab. Die Interpretationsfunktion $I_{RS_{ST}}$ ist eine Teilmenge der Interpretationsfunktion I_{RS} aus der „undifferenzierten" SIG_{OS}-Struktur $A_{SIG_{OS}}$. Mit Hilfe von $I_{RS_{ST}}$ werden alle Relationssymbole aus RS_{ST} extensional durch Mitglieder der Relationsfamilie RF_{ST} interpretiert.

Da die Extensionen dynamischer Relationssymbole per definitionem zeitlich variabel sind, können sie nicht ohne weiteres angegeben werden. Für die extensionale Interpretation von dynamischen Relationssymbolen ist es stets notwendig, einen „Bezeichner" für den Zustand anzugeben, in dem die Extension gilt. Denn die Extensionen von Relationssymbolen können sich zustandsspezifisch ändern. Daher muss die extensionale Interpretation einer dynamischen ontologischen Signatur SIG_{DY} stets bezogen auf einen Zustand z formuliert werden. Eine zustandsbezogene SIG_{DY}-Struktur A_{SIG_z} ist definiert als:

$$A_{SIG_z}=(OBF_{OS},OPF,RF_z,I_z)$$
$$\text{und } I_z=(I_K,I_{OPS},I_{RS_z}).$$

In einer zustandsbezogenen SIG_{DY}-Struktur A_{SIG_z} ist die gleiche Familie OBF_{OS} aller konzeptspezifischen Objektmengen enthalten, die auch in der statischen SIG_{ST}-Struktur $A_{SIG_{ST}}$ enthalten ist. Denn die Extensionen von Konzepten werden als zeitlich invariant angenommen. Ebenso ist auch die Familie OPF aller Operationen identisch. Hinsichtlich der Extensionen von Relationssymbolen wird allerdings die Zustandsbezogenheit von A_{SIG_z} deutlich. Denn die Extensionen von Relationssymbolen werden in Bezug auf den zugrunde gelegten Zustand z bestimmt. In dem Zustand z entspricht die Extension eines dynamischen Relationssymbols $R_j \in RS_{DY}$ derjenigen Relation $r_j \in RF_z$, auf die sie von der zustandsbezogenen Interpretationsfunktion

$$I_{RS_z}: RS_{DY} \rightarrow RF_z$$

abgebildet wird.

Mit Hilfe zustandsbezogener Strukturen zur Interpretation von Ontologien ist es nunmehr möglich, die zeitliche Varianz von Relationssymbolextensionen zu berücksichtigen. Beispielsweise kann es sein, dass ein dynamisches Relationssymbol R_j mit Hilfe einer ersten zustandsspezifischen Interpretationsfunktion $I_{RS_{z.1}}$ auf eine Relation r_{j_1} abgebildet wird, in der ein Objekttupel $(ob_1,...,ob_n)$ nicht enthalten ist. In einem zweiten Zustand z_2 kann hingegen die extensionale zustandsspezifische Interpretation $I_{RS_{z.2}}(R_j)=r_{j_2}$ u.a. auch $(ob_1,...,ob_n)$ umfassen. Mit dem Übergang von z_1 zu z_2 wird eine *Vergrößerung* der Extension von Relationssymbolen verdeutlicht, wenn ansonsten keine Änderung in r_{j_2} gegenüber r_{j_1} vorliegt. Mit dem Übergang von z_2 zu z_1 wird hingegen eine *Verkleinerung* der Relationssymbolextension verdeutlicht.

Die Angabe zustandsspezifischer Strukturen zur extensionalen Interpretation von Ontologien ist bereits ein erster Ansatz zur Erweiterung von Ontologien um dynamische Aspekte. Es können nämlich nunmehr zustandsspezifische Extensionen von Relationssymbolen berücksichtigt werden. Dennoch reicht eine solche „Parametrisierung“ von Strukturen mit Zuständen noch bei weitem nicht aus, um das Vorhaben der Erweiterung von Ontologien um dynamische Aspekte zu erfüllen. Denn mit der Angabe zustandsspezifischer Strukturen wird noch kein Zusammenhang zwischen den einzelnen Zuständen verdeutlicht. Insbesondere können – bei Beschränkung auf die bisherigen Ausdrucksmittel – keine *kausalen Zusammenhänge* zwischen Zuständen ausgedrückt werden. Solche kausalen Zusammenhänge liegen in der Durchführung von Operationen. Die Repräsentierbarkeit von operationsbedingten Übergängen von einer zustandsspezifischen Struktur zu einer zweiten zustandsspezifischen Struktur ist allerdings von wesentlichem Interesse bei der Modellierung dynamischer Phänomene.

3.1.4.3.3.2 Positive und negative Relationssymbole

Das Wissen eines Akteurs bezüglich einer atomaren ontologischen Formel[1) $R_j(t_1,...,t_n) \in FORM_{SIG_{OS}}$ kann auf zweifache Weise charakterisiert werden. Entweder der Akteur verfügt über das Wissen bezüglich des Gültigkeitsstatus der atomaren ontologischen Formel $R_j(t_1,...,t_n)$ in einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ oder der Akteur verfügt nicht über das Wissen bezüglich ihren Gültigkeitsstatus. Wenn der Akteur über das Wissen bezüglich des Gültigkeitssatus verfügt, kann er bestimmen, ob entweder die Formel $R_j(t_1,...,t_n)$ in $A_{SIG_{OS}}$ gültig ist ($A_{SIG_{OS}} \vDash R_j(t_1,...,t_n)$), oder in $A_{SIG_{OS}}$ ungültig ist ($A_{SIG_{OS}} \nvDash R_j(t_1,...,t_n)$). Verfügt der Akteur über Wissen bezüglich des Gültigkeitsstatus der Formel $R_j(t_1,...,t_n)$ in einer SIG_{OS}-Struktur $A_{SIG_{OS}}$, so verfügt er auch über Wissen bezüglich des Gültigkeitsstatus der Formel $\neg R_j(t_1,...,t_n)$ in $A_{SIG_{OS}}$. Wenn der Akteur hingegen nicht über das Wissen bezüglich des Gültigkeitsstatus von $R_j(t_1,...,t_n)$ in $A_{SIG_{OS}}$ verfügt, kann er weder über ihre Gültigkeit noch über ihre Ungültigkeit Aussagen machen.

In einem Petri-Netz kann lediglich Wissen über die Gültigkeit von Formeln repräsentiert werden. „Fehlende Aussagen" können als *Unwissen* bezüglich der Gültigkeit der entsprechenden Formel interpretiert werden. Wenn dieser Interpretation gefolgt wird, so werden von den Wissensausprägungen *Gültig – Ungültig – Unbekannt* lediglich die Ausprägungen *Gültig* und *Unbekannt* erfasst. Für die Wissensausprägung *Ungültig* wird in diesem Fall kein Ausdrucksmittel vorgesehen. Entsprechend dieser *Explizitheitsprämisse*[2) von Petri-Netzen entsprechen die stellenbezogenen Markierungen von höheren Petri-Netzen den Extensionen von Relationssymbolen.

Die Ungültigkeit von Formeln kann mit dem bislang vorgestellten formalen Apparat in Petri-Netzen nur durch eine zusammengesetzte Formel repräsentiert werden, die der Negation einer atomaren Formel entspricht. Diese Diskrepanz zwischen prädikatenlogischer und Petri-Netz-gestützter Wissensrepräsentation wird oftmals dadurch überbrückt, dass die Ungültigkeit aller Formeln angenommen wird, deren Gültigkeit nicht explizit aus der Netz-Markierung hervorgeht.[3) Mit Ontologie-Netzen ist hingegen – im Vergleich zu konventionellen Petri-Netzen – eine differenziertere Wissensrepräsentation möglich.[4) Es wird nämlich explizit zwischen den drei genannten Wissensausprägungen unterschieden. Zu diesem Zweck wird die Menge RS der Relationssymbole aus einer ontologischen Signatur SIG_{OS} derart erweitert, dass auch die Ungültigkeit einer atomaren ontologischen Formel durch eine wiederum atomare ontologische Formel repräsentiert werden kann.

Um Wissen über die Ungültigkeit einer ontologischen Formel $R_j(t_1,...,t_n)$ ausdrücken zu können, bieten sich zwei Alternativen an. Bei der ersten Alternative wird das Unwissen

1) Die folgenden Ausführungen lassen sich ebenso auf zusammengesetzte Formeln übertragen. Aus Gründen der Einfachheit werden jedoch lediglich atomare ontologische Formeln thematisiert.

2) Vgl. ZELEWSKI (1995), Bd. 4, S. 208.

3) Vgl. DAHR (1994), S. 17 ff., und die beispielhaften Spezifikationen in MESSER (1999), S. 118.

4) Vgl. Abschnitt 4.2.2.1.

bezüglich der Gültigkeit der Formel $R_j(t_1,...,t_n)$ als Nachweis von deren Ungültigkeit gewertet. Das heißt, dass auf $A_{SIG_{OS}} \not\models R_j(t_1,...,t_n)$ geschlossen wird, wenn kein Wissen darüber vorliegt, ob $A_{SIG_{OS}} \models R_j(t_1,...,t_n)$ gilt. Um die Ungültigkeit einer Formel $R_j(t_1,...,t_n)$ ausdrücken zu können, reicht es hierbei aus, sie nicht in die Faktenbasis aufzunehmen. Ein Kalkül, das solche Schlussfolgerungen erlaubt, wird im Rahmen der logischen Programmierung von dem *Negation-by-failure-Prinzip* erfasst.[1)] Das Negation-by-failure-Prinzip basiert auf der *Closed-world-Assumption*, der zufolge lediglich explizit angegebene Fakten Gültigkeit haben. Formeln, deren Gültigkeit nicht explizit angegeben ist, werden hingegen als ungültig angenommen.

Bei der zweiten Alternative wird für ein Relationssymbol $R_j \in RS$ ein zweites Relationssymbol $\underline{R}_j$ eingeführt, das als *negatives* Relationssymbol bezeichnet wird.[2)] Ein solches negatives Relationssymbol erlaubt es, atomare ontologische Formeln der Art $\underline{R}_j(t_1,...,t_n)$ zu konstruieren, mit denen die Ungültigkeit der ontologischen Formel $R_j(t_1,...,t_n)$ und somit die Gültigkeit der zusammengesetzten ontologischen Formel $\neg R_j(t_1,...,t_n)$ ausgedrückt werden kann.

Aus den beiden Alternativen, die für die Repräsentation von ungültigen Formeln in Frage kommen, entscheidet sich der Verfasser zu Gunsten der zweiten. Lediglich mit der zweiten Alternative ist es nämlich möglich, alle drei Ausprägungsformen des Wissens eines Akteurs explizit zu berücksichtigen. Bei der ersten Alternative wird hingegen die Unwissenheit des Akteurs bezüglich der Gültigkeit von Formeln mit seinem Wissen über deren Ungültigkeit vermengt.

Veränderungen der Faktenbasis, die auf Lern- oder Verlernprozesse des Akteurs zurückzuführen sind, können zudem bei der ersten Alternative nicht repräsentiert werden. Ob die Annahme über die Ungültigkeit einer ontologischen Formel auf ihre „faktische" Ungültigkeit oder auf das Unwissen des Akteurs zurückzuführen ist, kann nicht nachvollzogen werden. Darüber hinaus sind einerseits mit der Closed-world-Assumption potenziell fehlerhafte Schlussfolgerungen verbunden.[3)] Andererseits verhält sich die Annahme einer „geschlossenen Welt" konfliktionär mit den intendierten Anwendungen von Ontologien. Als intendierte Anwendungen kommen vordergründig Anwendungen im Rahmen des *Semantic Web* in Frage, die sich oftmals durch das Unwissen bezüglich des Zustands von Systemen auszeichnen. Mit Hilfe der expliziten Angabe des Unwissens bezüglich der Zustände von Systemen kann die Gefahr der fehlerhaften Verarbeitung von Informationen reduziert werden.

---

1) Vgl. SCHMID/KINDSMÜLLER (1996), S. 318.

2) Zu negativen Relationssymbolen vgl. DAHR (1994), S. 17; LI (1994), S. 384; ZELEWSKI (1995), Bd. 5.1, S. 164. Darüber hinaus findet sich in HE ET AL. (2003), S. 666, ein Hinweis auf negative Relationssymbole, demnach allerdings negative Relationssymbole „Synonyme" zu ihren positiven Pendants wären. Aus den späteren Ausführungen im Rahmen einer beispielhaften Formalisierung (vgl. HE ET AL. (2003), S. 667) kann jedoch auf einen Fehler bei der Einführung negativer Relationssymbole geschlossen werden.

3) Vgl. ZELEWSKI (1995), Bd. 4, S. 208 ff.

Die Menge $\underline{RS}$ umfasst alle negativen Relationssymbole aus einer ontologischen Signatur SIG_{OS}. Jedes negative Relationssymbol $\underline{R}_j \in \underline{RS}$ geht eineindeutig aus einem positiven Relationssymbol $R_j \in RS$ hervor.[1] Aus Gründen der Einfachheit wird für die Typisierung negativer Relationssymbole die Relationssymbol-Typisierungsfunktion $typ_{RS_{OS}}$ beibehalten. Der Typ $typ_{RS_{OS}}(\underline{R}_j)$ eines negativen Relationssymbols R_j stimmt dabei mit dem Typen $typ_{RS_{OS}}(R_j)$ des Relationssymbols überein, aus dem $\underline{R}_j$ abgeleitet wurde:

$$\forall R_j \in RS,\ \underline{R}_j \in \underline{RS}:$$
$$typ_{RS_{OS}}(R_j) = typ_{RS_{OS}}(\underline{R}_j).$$

Die Menge $FORM_{SIG_{OS}}$ aller ontologischen Formeln wird zudem derart erweitert, dass genau dann, wenn $R_j(t_1,...,t_n) \in FORM_{SIG_{OS}}$ gilt, auch $\underline{R}_j(t_1,...,t_n) \in FORM_{SIG_{OS}}$ gilt:

$$\forall R_j \in RS, t_1,...,t_n \in TERM_{SIG_{OS}}: R_j(t_1,...,t_n) \in FORM_{SIG_{OS}} \leftrightarrow \underline{R}_j(t_1,...,t_n) \in FORM_{SIG_{OS}}.$$

Eine ontologische Formel $\underline{R}_j(t_1,...,t_n)$ wird genau dann von einer Variablenbelegung $belf_{OS}$ in einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ bestätigt, wenn die ontologische Formeln $R_j(t_1,...,t_n)$ von $belf_{OS}$ in $A_{SIG_{OS}}$ nicht bestätigt wird. Entsprechend ist $\underline{R}_j(t_1,...,t_n)$ genau dann in $A_{SIG_{OS}}$ gültig, wenn $R_j(t_1,...,t_n)$ in $A_{SIG_{OS}}$ ungültig ist. Die ontologische Formel $\underline{R}_j(t_1,...,t_n)$ ist genau dann in $A_{SIG_{OS}}$ ungültig, wenn $R_j(t_1,...,t_n)$ in $A_{SIG_{OS}}$ gültig ist (vice versa). Somit gilt die *Äquivalenz*:

$$\forall R_j(t_1,...,t_n), \underline{R}_j(t_1,...,t_n) \in FORM_{SIG_{OS}}:$$
$$\neg R_j(t_1,...,t_n) \equiv \underline{R}_j(t_1,...,t_n).$$

Entsprechend gilt auch:

$$\forall R_j(t_1,...,t_n), \underline{R}_j(t_1,...,t_n) \in FORM_{SIG_{OS}}:$$
$$(A_{SIG_{OS}} \vDash \underline{R}_j(t_1,...,t_n)) \leftrightarrow (A_{SIG_{OS}} \nvDash R_j(t_1,...,t_n)).$$

Die „doppelte Negation“ $\underline{\underline{R}}_j$ eines Relationssymbols R_j stimmt mit R_j überein. Somit gilt die Äquivalenz $R_j(t_1,...,t_n)) \equiv \underline{\underline{R}}_j(t_1,...,t_n)$. Das Relationssymbol $\underline{\underline{R}}_j$ kann sowohl als „doppelte Negation“ des Relationssymbols R_j als auch als „einfache Negation“ des negativen Relationssymbols $\underline{R}_j$ charakterisiert werden. Daher wird darauf verzichtet, die Formelmenge $FORM_{SIG_{OS}}$ um solche Formeln zu erweitern, die mit doppelt negierten Relationssymbolen konstruiert sind.

Die extensionale Interpretation $\underline{r}_j$ eines negativen Relationssymbols $\underline{R}_j \in \underline{RS}$ mit $typ_{RS_{OS}}(\underline{R}_j) = (k_1,...,k_n)$ lautet:

$$\underline{r}_j = \{ (ob_1,...,ob_n) \mid (ob_1,...,ob_n) \in (OB_{k_1} \times ... \times OB_{k_n}) \wedge (ob_1,...,ob_n) \notin r_j \}.$$

Die Extension $\underline{r}_j$ eines negativen Relationssymbols $\underline{R}_j$ zu einem positiven Relationssymbol R_j mit $I_{RS}(R_j) = r_j$ umfasst demnach alle Objekttupel der Art $(ob_1,...,ob_n)$, die in dem

1) Somit kann die typographische Unterstreichung der positiven Relationssymbole als die Anwendung einer bijektiven Funktion interpretiert werden. Anstelle der Unterstreichung bietet es sich auch an, das zu einem positiven Relationssymbol $R_j \in RS$ zugehörige negative Relationssymbol als $neg(R_j)$ anzugeben. Durch diese Notation wird allerdings die Lesbarkeit von Formeln mit Hilfe negativer Relationssymbole erheblich erschwert.

kartesischen Produkt $OB_{k_1}\times...\times OB_{k_n}$, aber nicht in r_j enthalten sind. Die Extension r_j zum positiven Relationssymbol R_j ist als Teilmenge von $OB_{k_1}\times...\times OB_{k_n}$ bestimmt. Somit lässt sich die Extension $\underline{r}_j$ zu einem negativen Relationssymbol $\underline{R}_j$ auch als

$$\underline{r}_j = (OB_{k_1}\times...\times OB_{k_n}) \setminus r_j$$

angeben. Somit darf es kein Objekttupel $(ob_1,...,ob_n)\in(OB_{k_1}\times...\times OB_{k_n})$ geben, das sowohl in der Extension r_j zum positiven Relationssymbol R_j als auch in der Extension $\underline{r}_j$ zum negativen Relationssymbol $\underline{R}_j$ enthalten ist.[1)]

Die Differenzierungen von Relationssymbolen in einerseits statische und dynamische Relationssymbole sowie andererseits positive und negative Relationssymbole können miteinanander kombiniert werden. Relationssymbole aus einer ontologischen Signatur SIG_{OS} lassen sich entsprechend den vier Kombinationsmöglichkeiten für die Kriterien *Dynamik* und *Vorzeichen* in einer Vier-Felder-Matrix zusammenen. In Tabelle 10 sind die Kombinationsmöglichkeiten für die Eigenarten von Relationssymbolen wiedergegeben.

		Vorzeichen	
		negativ	positiv
Dynamik	statisch	$\underline{R}_j\in\underline{RS_{ST}}$	$R_j\in RS_{ST}$
	dynamisch	$\underline{R}_j\in\underline{RS}_{DY}$	$R_j\in RS_{DY}$

Tabelle 10: Kombinationsmöglichkeiten für Eigenarten von Relationssymbolen

1) Diese Feststellung ist insofern von Bedeutung, als dass es sie in Ontologie-Netzen mit Integritätstransitionen abzusichern gilt. Integritätstransitionen werden in Abschnitt 4.2.2.1.2.2 thematisiert.

3.2 Petri-Netze

3.2.1 Allgemeine Petri-Netze

Petri-Netze sind – neben Ontologien – der zweite Baustein im integrativen Modellierungsansatz. Die operationale Semantik von Ontologien wird nämlich im Rahmen des integrativen Modellierungskonzepts mit der dynamischen Struktur von *Ontologie-Netzen* identifiziert. Letztgenannte werden wiederum als Erweiterung von bereits vorliegenden Petri-Netzen-Klassen eingeführt. Daher werden in den folgenden Abschnitten Petri-Netze mit steigender Komplexität vorgestellt. Während allerdings Ontologien noch ein sehr junges Forschungsgebiet darstellen, kann im Fall von Petri-Netzen auf ein reichhaltiges theoretisches Fundament zurückgegriffen werden. Es würde den Rahmen dieser Arbeit sprengen, sämtliche Aspekte zu berücksichtigen, die im Umfeld von Petri-Netzen in den letzten Jahren erarbeitet wurden. Daher wird im Folgenden lediglich auf jene Aspekte eingegangen, die eine unmittelbare Relevanz für die Modellierung kooperativer Informationssysteme haben. Für weiterführende Aspekte wird auf die entsprechende Literatur verwiesen.

In allgemeinen Petri-Netzen werden die notwendigen Bestandteile jeder weiteren Petri-Netz-Klasse definiert. Daher werden sie auch als Petri-Netze i.e.S. bezeichnet.[1)] In einem weiter gefassten Verständnis umfasst die extensionale Interpretation des Begriffs *Petri-Netz* formale Objekte, auf die alle Eigenschaften allgemeiner Petri-Netze zutreffen. Der üblichen Differenzierung folgend können allgemeine Petri-Netze in *elementare* und *höhere* Petri-Netze unterteilt werden. Sowohl elementare als auch höhere Petri-Netze weisen alle Bestandteile von allgemeinen Petri-Netzen auf. Die taxonomischen Beziehungen zwischen den Petri-Netz-Klassen entsprechen der Darstellung in Abbildung 14.

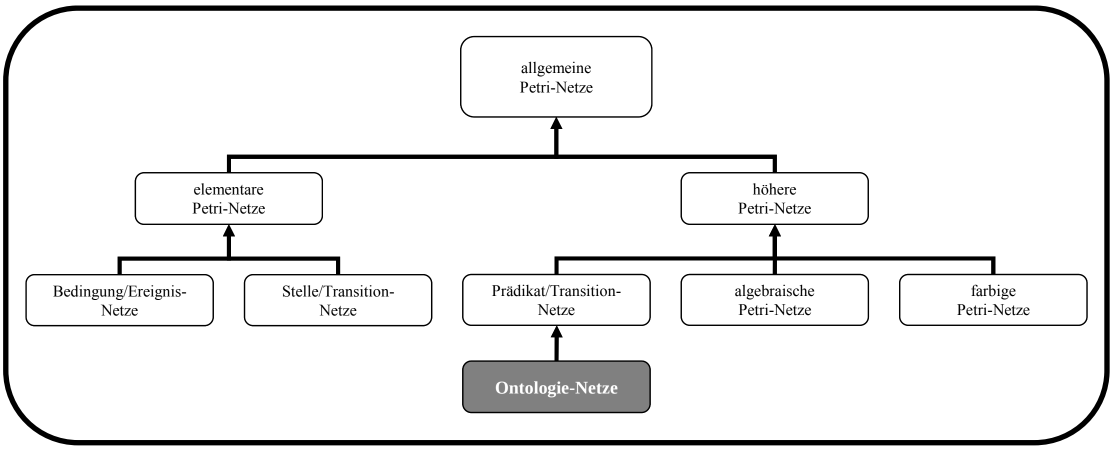

Abbildung 14: Petri-Netz-Taxonomie

1) Vgl. ZELEWSKI (1995), Bd. 3, S. 25.

Die Abbildung 14 gibt einen groben Überblick über jene Petri-Netz-Klassen, die im Rahmen des Petri-Netz-Konzepts die weiteste Verbreitung aufweisen.[1)] Darüber hinaus sind – zumeist aufbauend auf den o.a. „Basisklassen“ – vielfach sich voneinander unterscheidende Petri-Netz-Klassen entwickelt worden. Zumeist sind die „Spielarten“ des Petri-Netz-Konzepts aus einer spezifischen Zweckorientierung heraus entwickelt worden.

In dieser Arbeit wird eine Klasse höherer Petri-Netze vorgestellt, die sich primär als Unterklasse von Prädikat/Transition-Netzen einordnen lässt. Die Ausprägungen der neu vorzustellenden Petri-Netz-Klasse werden als *Ontologie-Netze* bezeichnet, da einerseits für die Annotation der Komponenten des Petri-Netzes Komponenten aus einer Ontologie $SPEZ_{OS}$ herangezogen werden und andererseits für Ontologie-Netze auch ein Instrumentarium vorhanden ist, das es erlaubt, Inferenz- und Integritätsregeln in Teilnetze zu überführen. Darüber hinaus weisen Ontologie-Netze hinsichtlich der artspezifischen Differenzierung von Termen auch einen Bezug zu algebraischen Petri-Netzen auf.

Zwecks einfacherer Diktion werden im Folgenden zunächst konstitutive Merkmale von allgemeinen und – darauf aufbauend – von Stelle/Transition-Netzen vorgestellt.

1) Zu Überblicken über Petri-Netz-Klassen vgl. BOBEANU ET AL. (2004), S. 392 f.; WEBER (2002), S. 14 ff.

Ein allgemeines Petri-Netz PN wird definiert durch das Drei-Tupel[1)]:

$$PN=(ST,TR,FR)$$

mit
IB_{PN_1}: $ST \cap TR=\varnothing$,
IB_{PN_2}: $(ST \cup TR) \neq \varnothing$ und
IB_{PN_3}: $(ST \cup TR) = (VB_{FR}(FR) \cup NB_{FR}(FR))$.

Die Komponenten eines allgemeinen Petri-Netzes PN sind:

(1.) eine Menge $ST=\{st_m\}$ von Stellen mit $m\in\{1,...,M\}$ und $M\in\mathbb{N}_+$,

(2.) eine Menge $TR=\{tr_n\}$ von Transitionen mit $n\in\{1,...,N\}$ und $N\in\mathbb{N}_+$ und

(3.) eine *Flussrelation* FR mit $FR\subseteq((ST\times TR) \cup (TR\times ST))$.

Die Mengen ST und TR werden gemeinsam auch als *Knotenmengen* bezeichnet. Die Elemente der Menge $KN=(ST\cup TR)$ werden entsprechend als *Knoten* bezeichnet. Die Teilmengen ST und TR der Knotenmenge KN müssen beide jeweils endlich sein. Unendlich große Mengen von Stellen und Transitionen werden daher ausgeschlossen.

Die Elemente $(kn_{x_1},kn_{y_1}),...,(kn_{x_n},kn_{y_n})$ der Flussrelation FR werden als (gerichtete) *Flusskanten* bezeichnet. Alternativ wird auch die Bezeichnung *Kante* zugelassen, wenn aus dem Kontext hervorgeht, dass es sich um ein Element der Flussrelation FR handelt.[2)] Der Knoten $kn_x\in(ST\cup TR)$, von dem eine Kante $(kn_x,kn_y)\in FR$ ausgeht, wird als *Ursprungsknoten* der Kante (kn_x,kn_y) bezeichnet. Der *Zielknoten* der Kante (kn_x,kn_y) ist der Knoten $kn_y\in(ST\cup TR)$. Die Kante $(kn_x,kn_y)\in FR$ wird zudem als *Eingangskante* des Zielknotens kn_y und als *Ausgangskante* des Ursprungsknotens kn_x bezeichnet.

Petri-Netze lassen sich als *bipartite Graphen*[3)] einordnen. Die Bipartitheit von Petri-Netzen ergibt sich aus den *Integritätsbedingungen* IB_{PN_1} und der Definition (3.) der Flussrelation FR. Es handelt es sich bei IB_{PN_1} um die *Disjunktheitsbedingung* $ST\cap TR=\varnothing$.[4)] Aufgrund der Disjunktheitsbedingung darf kein formales Objekt $kn\in(ST\cup TR)$ sowohl als Stelle als auch als Transition ausgezeichnet sein. Die Elemente

1) Vgl. BAUMGARTEN (1996), S. 50; KINDLER (1995), S. 26; REISIG (1991), S. 17; ROZENBERG/ENGELFRIET (1998), S. 18; ZELEWSKI (1995), Bd. 3, S. 25.

2) Die Unterscheidung ist insofern von Bedeutung, als im weiteren Verlauf Informationskanten als weitere Ausprägungsform von Kanten vorgestellt werden.

3) Bipartite Graphen sind Unterarten r-partiter Graphen. In r-partiten Graphen (KN,KA) kann die Kotenmenge KN so in r zueinander disjunkte Teilmengen $KN=KN_1\cup...\cup KN_r$ mit $KN_x\cap KN_y=\varnothing$ für alle $x,y\in\{1...r\}$ und $x\neq y$ unterteilt werden, dass der Zielknoten jeder Kante aus K in einer anderen Menge liegt als der Ursprungsknoten der Kante; vgl. DIESTEL (2000), S. 15 f. In bipartiten Graphen ist entsprechend die Menge KN der Knoten so in zwei disjunkte Teilmengen unterteilt, dass jede Kante immer von einem Element aus der einen Menge zu einem Element aus der anderen Menge führen muss; vgl. BAUMGARTEN (1996), S. 43; BRANDSTÄDT (1994), S. 14; DIESTEL (2000), S. 15 f.

4) Vgl. ZELEWSKI (1995), Bd. 3, S. 30.

der Menge ST aller Stellen oder der Menge TR aller Transitionen weisen nämlich jeweils unterschiedliche Intensionen auf. Wie später noch aufgezeigt wird, korrespondieren nämlich Stellen mit *passiven* Elementen, während Transitionen *aktiven* Elementen entsprechen. Die Dichotomisierung aktiv-passiv lässt es nicht zu, dass ein formales Objekt kn sowohl als aktives als auch als passives Element ausgezeichnet wird. Zweitens wird mit der Definition der Flussrelation FR als Teilmenge der Menge $((ST \times TR) \cup (TR \times ST))$ gewährleistet, dass nur gerichtete Kanten zwischen artverschiedenen Knoten zugelassen sind. Aus der Dichotomisierung der Knotenmenge und der Definition (3.) der Flussrelation ergibt sich die bipartite Struktur allgemeiner Petri-Netze.

Bei der zweiten Integritätsbedingung $(ST \cup TR) \neq \varnothing$ handelt es sich um die *Existenzbedingung*. Demnach muss mindestens eine Stelle $st_m \in ST$ oder mindestens eine Transition $tr_n \in TR$ existieren, damit ein allgemeines Petri-Netz PN vorliegen kann.

Die dritte Integritätsbedingung $(ST \cup TR) = (VB_{FR}(FR) \cup NB_{FR}(FR))$ wird als *Verknüpftheitsbedingung* bezeichnet.[1] Für die Formulierung der Verknüpftheitsbedingung ist die Bestimmung des *Vorbereichs* $VB_{FR}(FR)$ und des *Nachbereichs* $NB_{FR}(FR)$ der Flussrelation FR notwendig:[2]

$$VB_{FR}(FR) = \{kn_x \mid kn_x \in (ST \cup TR) \land \exists kn_y \in (ST \cup TR): (kn_x, kn_y) \in FR).$$

Der Vorbereich $VB_{FR}(FR)$ der Flussrelation FR umfasst alle Knoten $kn_x(ST \cup TR)$, die Eingangsknoten zu mindestens einem weiteren Knoten $kn_y \in (ST \cup TR)$ sind. Analog dazu ist der Nachbereich $NB_{FR}(FR)$ zu der Flussrelation FR definiert als:

$$NB_{FR}(FR) = \{kn_y \mid kn_y \in (ST \cup TR) \land \exists kn_x \in (ST \cup TR): (kn_x, kn_y) \in FR).$$

De Nachbereich $NB_{FR}(FR)$ der Flussrelation FR umfasst alle Knoten $kn_y \in (ST \cup TR)$, die Ausgangsknoten zu mindestens einem weiteren Knoten $kn_y \in (ST \cup TR)$ sind.

Die Verknüpftheitsbedingung fordert demnach, dass jeder Knoten $kn \in ST \cup TR$ im Petri-Netz mindestens einmal im Vor- oder Nachbereich der Flussrelation FR vorkommt. Würde die Verknüpftheitsbedingung nicht gelten, könnten auch *isolierte* Knoten im Petri-Netz existieren, die in keinem Paar $(kn_x, kn_y) \in FR$ vorkommen.

In Verbindung mit der Existenzbedingung hat die Verknüpftheitsbedingung zur Folge, dass die kleinstmöglichen Petri-Netze als Grenzfälle wohldefiniert sind.[3] Wenn nämlich in einem Petri-Netz PN mindestens ein Knoten kn_x vorkommen muss (Existenzbedingung), der zudem in mindestens einem Knotenpaar vorkommen muss (Verknüpftheitsbedingung), dann muss es auch mindestens einen zweiten Knoten kn_y von der jeweils anderen Knotenart (Disjunktheitsbedingung) geben, der Eingangs- oder Ausgangsknoten von kn_x ist. Der erste Grenzfall $PN_1 = (\{st_1\}, \{tr_1\}, \{(tr_1, st_1)\})$ umfasst eine einelementige Menge ST, eine einelementige Menge TR und eine einelementige Menge FR mit

1) Vgl. ZELEWSKI (1995), Bd. 3, S. 25.

2) Vgl. ZELEWSKI (1995), Bd. 3, S. 11.

3) Vgl. ZELEWSKI (1995), Bd. 3, S. 26 f.

$FR=\{(tr_1,st_1)\}$. In diesem Fall ist die einzige Kante (tr_1,st_1) im Petri-Netz PN_1 von der einzigen Transition tr_1 zu der einzigen Stelle st_1 definiert. Der zweite Grenzfall $PN_2=(\{st_1\},\{tr_1\},\{(st_1,tr_1)\})$ umfasst auch eine einelementige Menge ST und eine einelementige Menge TR. Die einelementige Menge FR ist hingegen als $FR=\{(st_1,tr_1)\}$ definiert. Die einzige gerichtete Kante (st_1,tr_1) führt in diesem Fall von der einzigen Stelle st_1 zu der einzigen Transition tr_1.

Analog zu der Bestimmung des Vor- und Nachbereichs der Flussrelation FR können die Vor- und Nachbereiche von Stellen oder Transitionen bestimmt werden. Der Vorbereich $VB_{ST}(st_m)$ bzw. der Nachbereich $NB_{ST}(st_m)$ zu einer Stelle $st_m \in ST$ ist bestimmt als:[1)]

$$VB_{ST}(st_m)=\{tr_n \mid tr_n \in TR \wedge (tr_n,st_m) \in FR\}$$
$$NB_{ST}(st_m)=\{tr_n \mid tr_n \in TR \wedge (st_m,tr_n) \in FR\}.$$

Die Menge

$$NA_{ST}(st_m)=VB_{ST}(st_m) \cup NB_{ST}(st_m)$$

ist die *Nachbarschaft* der Stelle st_m. Die Nachbarschaft $NA_{ST}(st_m)$ einer Stelle st_m umfasst alle Transitionen aus ihrem Vor- und ihrem Nachbereich.

Analog sind der Vorbereich $VB_{TR}(tr_n)$, der Nachbereich $NB_{TR}(tr_n)$ und die Nachbarschaft $NA_{TR}(tr_n)$ zu einer Transition $tr_n \in TR$ definiert als:

$$VB_{TR}(tr_n)=\{st_m \mid st_m \in TR \wedge (st_m,tr_n) \in FR\}$$
$$NB_{TR}(tr_n)=\{st_m \mid st_m \in TR \wedge (tr_n,st_m) \in FR\}$$
$$NA_{TR}(tr_n)=VB_{TR}(tr_n) \cup NB_{TR}(tr_n).$$

Der Vorbereich $VB_{KN}(kn_x)$, der Nachbereich $NB_{KN}(kn_x)$ und die Nachbarschaft $NA_{KN}(kn_x)$ eines artspezifisch nicht näher bestimmten Knotens $kn_x \in ST \cup TR$ sind definiert als:

$$VB_{KN}(kn_x)=\{kn_y \mid kn_y \in ST \cup TR \wedge (kn_y,kn_x) \in FR\}$$
$$NB_{KN}(kn_x)=\{kn_y \mid kn_y \in ST \cup TR \wedge (kn_x,kn_y) \in FR\}$$
$$NA_{KN}(kn_x)=VB_{KN}(kn_x) \cup NB_{KN}(kn_y).$$

Die Menge $VB_{KN}(kn)$ zu einem Knoten $kn \in (ST \cup TR)$ wird auch als dessen *Eingangsknotenmenge* bezeichnet. Entsprechend wird die Menge $NB_{KN}(kn)$ zu einem Knoten $kn_x \in ST \cup TR$ auch als dessen *Ausgangsknotenmenge* bezeichnet. Wenn in der Eingangsknotenmenge $VB_{KN}(kn)$ eines Knotens kn_x mindestens zwei weitere Knoten $kn_y,kn_z \in (ST \cup TR)$ enthalten sind, so wird kn_x als *rückwärtsverzweigt* bezeichnet[2)]. Wenn in der Ausgangsknotenmenge $NB_{KN}(kn)$ eines Knotens kn_x mindestens zwei weitere Knoten $kn_y,kn_z \in (ST \cup TR)$ enthalten sind, so wird kn_x als *vorwärtsverzweigt* bezeichnet[3)].

1) Vgl. ZELEWSKI (1995), Bd. 3, S. 11.

2) Vgl. BAUMGARTEN (1996), S. 51.

3) Vgl. BAUMGARTEN (1996), S. 51.

Ein Knoten $kn_y \in (ST \cup TR)$ wird als *inzident* zu einem Knoten $kn_x(ST \cup TR)$ bezeichnet, wenn die Kante (kn_x,kn_y) oder (kn_y,kn_x) in der Flussrelation FR enthalten ist. Jeder zu einem Knoten kn_x inzidente Knoten kn_y ist im Vorbereich $VB_{KN}(kn_x)$ oder im Nachbereich $NB_{KN}(kn_x)$ von kn_x enthalten. Bei der Inzidenz von zwei Knoten handelt es sich um eine symmetrische Beziehung. Wenn ein Knoten $kn_x \in (ST \cup TR)$ zu einem Knoten $kn_y \in (ST \cup TR)$ inzident ist, so ist auch kn_y zu kn_x inzident. Zueinander inzidente Knoten sind wechselseitig in den jeweiligen Vor- und Nachbereichen enthalten.[1] Die Kante $(kn_x,kn_y) \in FR$, die zwei zueinander inzidente Knoten $kn_x,kn_y \in (ST \cup TR)$ miteinander verbindet, wird als *adjazent* zu beiden Knoten kn_x und kn_y bezeichnet. Die zueinander inzidenten Knoten kn_x und kn_y werden auch als adjazent zu der Kante $(kn_x,kn_y) \in FR$ bezeichnet.

Ein Petri-Netz PN wird als *schlicht* bezeichnet, wenn es in PN keine zwei Knoten $kn_x,kn_y \in (ST \cup TR)$ gibt, deren Eingangsknotenmengen $VB_{KN}(kn_x)$ bzw. $VB_{KN}(kn_y)$ und Ausgangsknotenmengen $NB_{KN}(kn_x)$ bzw. $NB_{KN}(kn_y)$ gleich sind[2]. Somit gilt für schlichte Petri-Netze:

$$\forall x,y \in ST \cup TR: (VB_{KN}(x)=VB_{KN}(y) \wedge NB_{KN}(x)=NB(y)) \rightarrow x=y.$$

Ein Petri-Netz $PN_2=(ST_2,TR_2,FR_2)$ ist ein *Teilnetz* eines weiteren Petri-Netzes $PN_1=(ST_1,TR_1,FR_1)$, wenn

$$ST_2 \subseteq ST_1,$$
$$TR_2 \subseteq TR_1 \text{ und}$$
$$FR_2 = FR_1 \cap ((ST_2 \times TR_2) \cup (TR_2 \times ST_2))$$

gelten.[3] Ein Teilnetz $PN_1=\{\{st_m\},\{tr_n\},\{(st_m,tr_n),(tr_n,st_m)\}$ eines Petri-Netzes $PN_2=(ST,TR,FR)$ mit $st_m \in ST$, $tr_n \in TR$ und $(st_m,tr_n),(tr_n,st_m) \in FR$ wird als *Schleife* bezeichnet. Schleifen zeichnen sich dadurch aus, dass der Vor- und Nachbereich $VB_{ST}(st_m)$ bzw. $NB_{ST}(st_m)$ der einzigen Stelle st_m zusammenfallen.[4] Die einzige Stelle st_m in PN_1 ist in dem Fall eine *Nebenbedingung* für die einzige Transition tr_n. Petri-Netze, die keine Schleifen enthalten, werden als *rein* bezeichnet.[5] Mit *unreinen* Petri-Netzen sind Probleme verbunden, die im nächsten Abschnitt thematisiert werden.

Die Definition von Petri-Netzen als bipartite Graphen kommt ihrer graphischen Darstellung zu Gute. Die Komponenten eines Petri-Netzes PN=(ST,TR,FR) lassen sich nämlich unmittelbar von der formalen Spezifikation in die graphische Darstellung übertragen. Dabei kann das Petri-Netz PN durch einen *visuellen Graphen* dargestellt werden,

1) Vgl. ZELEWSKI (1995), Bd. 3, S. 11.

2) Vgl. BAUMGARTEN (1996), S. 51.

3) Vgl. BAUMGARTEN (1996), S. 51.

4) Vgl. BAUMGARTEN (1996), S. 53; REISIG (1991), S. 17; STARKE (1990), S. 32.

5) Vgl. REISIG (1991), S. 17; STARKE (1990), S. 32.

der für jede Stelle $st_m \in ST$ einen *Kreis* und für jede Transition $tr_n \in TR$ ein *Rechteck*[1]) enthält. Die gerichteten Kanten (kn_x, kn_y) aus der Flussrelation FR werden durch Pfeile dargestellt, wobei die Pfeilspitze jeweils dem Zielknoten kn_y der Kante zugeordnet ist. Entsprechend hat jeder Pfeil, der im visualisierten Graphen ein Kante $(kn_x, kn_y) \in FR$ repräsentiert, seinen Ursprung in dem Symbol, das den Knoten kn_x repräsentiert.

Der visuelle Graph in Abbildung 15 ist ein Beispiel für die graphische Darstellung eines allgemeinen Petri-Netzes

$PN=(\{st_1,st_2\},\{tr_1,tr_2\},\{(st_1,tr_2),(st_2,tr_1),(tr_1,st_1),(tr_2,st_2)\})$

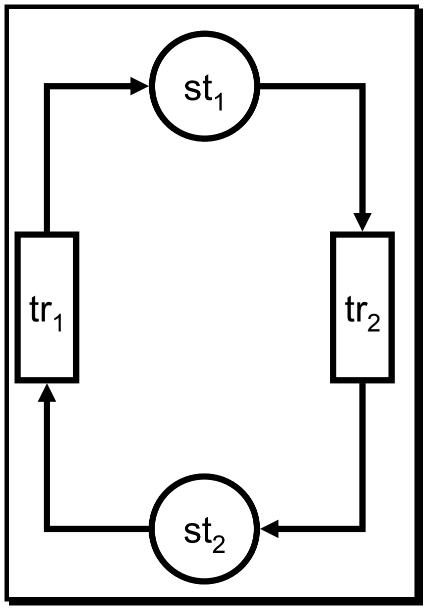

Abbildung 15: Allgemeines Petri-Netz PN

Die Komponenten des Petri-Netzes PN lauten:

$ST=\{st_1,st_2\}$,

$TR=\{tr_1,tr_2\}$ und

$FR=\{(st_1,tr_2),(st_2,tr_1),(tr_1,st_1),(tr_2,st_2)\}$.

---

1) Alternativ werden statt Rechtecken auch *Linien* verwendet; vgl. z.B. ADAM ET AL. (1998), S. 140 ff.; LIN/ CHANSON (1998), S. 827 ff. Von dieser Alternative wird allerdings abgesehen, da die Unterscheidung zwischen Kanten und Transitionen in der graphischen Darstellung dadurch erschwert wird. Darüber hinaus wurden von GRUHN/KAMPMANN *Funsoft-Netze* als Unterart allgemeiner Petri-Netze vorgestellt, für die ikonische Darstellungen mit intuitiver Semantik verwendet werden; vgl. GRUHN/KAMPMANN (1996), S. 384 ff. Stellen („Kanäle“) werden dort weiterhin als Kreise illustriert. Für Transitionen („Instanzen“) werden Symbole verwendet, die den repräsentierten Aktivitäten oder aktivitätsdurchführenden Akteuren oder benutzten Werkzeugen entsprechen.

3.2.2 Stelle/Transition-Netze

3.2.2.1 Definition von Stelle/Transition-Netzen

Allgemeine Petri-Netze sind die Grundlage für alle weiteren Petri-Netz-Klassen. Die Grundlage umfasst in erster Linie die Petri-Netz-Topologie[1]. Diese topologischen Eigenarten finden sich hauptsächlich in der Eigenart der Flussrelation FR wieder, nur zwei Knoten unterschiedlicher Knotenarten miteinander zu verknüpfen. Darüber hinaus wird die Netztopologie durch die Integritätsbedingungen IB_{PN_1}, IB_{PN_2} und IB_{PN_3} fixiert, die auch für Stelle-Transition-Netze (S/T-Netze) gültig sind.

Trotz der Gemeinsamkeiten mit allgemeinen Petri-Netzen weisen S/T-Netze einige bemerkenswerte Eigenarten auf. Insbesondere wird die Ausdrucksmächtigkeit des Petri-Netz-Konzepts um das „Markenspiel" erweitert. Als Baustein hierfür dient ein identitätsloses, unstrukturiertes formales Objekt, das als *Basismarke* bezeichnet wird.[2] Die Kopien dieser Basismarke werden vereinfacht als *Marken* bezeichnet. Bei Marken handelt es sich um formale Objekte, die sich von den bislang vorgestellten formalen Objekten, die in allgemeinen Petri-Netzen enthalten sind, in einem Punkt wesentlich unterscheiden. Im Gegensatz zu Stellen und Transitionen handelt es sich bei Marken um Objekte, die zustandsabhängig positioniert werden. Zu diesem Zweck wird das Petri-Netz-Konzept um eine *strukturelle Dynamik* angereichert.

Ein S/T-Netz STN ist definiert als:[3]

$$STN=(ST,TR,FR,KP,G,M_0)$$

mit

IB_{PN_1}: $(ST \cap TR) = \varnothing$,
IB_{PN_2}: $(ST \cup TR) \neq \varnothing$,
IB_{PN_3}: $(ST \cup TR) = (VB_{FR}(FR) \cup NB_{FR}(FR))$,
IB_{STN_1}: $\forall st_m \in ST,\ tr_n \in TR$: $G(tr_n,st_m) \leq KP(st_m) \geq G(st_m,tr_n)$ und
IB_{STN_2}: $\forall st_m \in ST$: $0 \leq M_0(st_m) \leq KP(st_m)$.

Die Komponenten eines S/T-Netzes STN sind:

---

1) Die *Topologie* eines Petri-Netzes ist die Gesamtheit bestehend aus der Knotenmenge ST, der Transitionenmenge TR, der Flussrelation FR mit $FR \subseteq ((ST \times TR) \cup (TR \times ST))$ sowie den Integritätsbedingungen IB_{PN_1}, IB_{PN_2} und IB_{PN_3}. Mit der Topologie eines Petri-Netzes wird die „systemische" Struktur des Petri-Netzes – d.h. seines Zusammenhangs aus Knoten (Elemente) und Kanten (Beziehungen) – angegeben, ohne auf irgendeine Weise auf metrische Eigenschaften von Netzkomponenten Bezug zu nehmen. Aufgrund der nicht-metrischen Beschreibung einer Netzstruktur wird die topologische Beschreibung einer Netzstruktur auch als „qualitative" Strukturbeschreibung charakterisiert.

2) Vgl. ZELEWSKI (1995), Bd. 3, Fn. 2 auf S. 7.

3) Vgl. BAUMGARTEN (1996), S. 79; DESEL/REISIG (1998), S. 129 ff.; GIRAULT/VALK (2003), S. 41 f.; REISIG (1991), S. 71; ZELEWSKI (1995), Bd. 3, S. 30.

(1.) eine Menge $ST=\{st_m\}$ von *Stellen* mit $m\in\{1,...,M\}$ und $M\in\mathbb{N}_+$,

(2.) eine Menge $TR=\{tr_n\}$von *Transitionen* mit $n\in\{1,...,N\}$ und $N\in\mathbb{N}_+$,

(3.) eine *Flussrelation* FR mit $FR\subseteq((ST\times TR)\cup(TR\times ST))$,

(4.) eine *Kapazitätsfunktion* $KP: ST \rightarrow \mathbb{N}_+\cup\{\infty\}$,

(5.) eine *Gewichtsfunktion* $G: ((ST \times TR) \cup (TR \times ST)) \rightarrow \mathbb{N}$ und

(6.) eine *Anfangsmarkierungsfunktion* $M_0: ST \rightarrow \mathbb{N}$.

Der Unterordnung von S/T-Netzen gegenüber allgemeinen Petri-Netzen folgend, gelten auch für alle S/T-Netze die Integritätsbedingungen der Disjunktheit (IB_{PN_1}), der Existenz (IB_{PN_2}) und der Verknüpftheit (IB_{PN_3}). Darüber hinaus wird für die Zulässigkeit von S/T-Netzen ihre Konformität mit den Integritätsbedingungen IB_{STN_1} und IB_{STN_2} eingefordert. Bezüglich der Integritätsbedingung IB_{STN_1} gelten solche S/T-Netze als unzulässig, in denen Stellen vorkommen, deren Kapazität kleiner ist als die Gewichtung einer ihrer Ausgangs- oder Eingangskanten. Bezüglich der Integritätsbedingung IB_{STN_2} sind hingegen alle S/T-Netze unzulässig, in denen eine stellenbezogene Markierung existiert, die negativ ist oder die Kapazität der entsprechenden Stelle übersteigt.

Über die Netztopologie hinaus sind in S/T-Netzen drei Komponenten enthalten, die für die *Struktur* von S/T-Netzen von wesentlicher Bedeutung sind. Die Erweiterungen hinsichtlich allgemeiner Petri-Netze liegen in der *Kapazitätsfunktion* KP, der *Gewichtungsfunktion* G und der Anfangsmarkierungsfunktion M_0.[1)] Zwecks Vorstellung dieser Struktur wird in den folgenden Abschnitten zwischen einerseits der *statischen* und andererseits der *dynamischen Struktur* von S/T-Netzen unterschieden.

Graphisch lassen sich S/T-Netze unter einer Markierung M_0 visualisieren, indem die Anzahl $M_0(st_m)$ der Marken, die durch die Markierungsfunktion M_0 für jede Stelle $st_m\in ST$ definiert ist, als Anzahl von Punkten auf den stellenrepräsentierenden Kreisen dargestellt wird. In der Abbildung 16 ist ein beispielhaftes S/T-Netz gegeben. Die Gewichtung $G(kn_x,kn_y)$ eines Knotenpaars $(kn_x,kn_y)\in((ST\times TR)\cup(TR\times ST))$, das in der Flussrelation FR enthalten ist, wird als Anschrift des entsprechenden Pfeils im visuellen Graphen eingezeichnet. Fehlende Pfeilanschriften entsprechen der Gewichtung $G(kn_x,kn_y)=1$. Beispielsweise kann das S/T-Netz

1) Die gegenüber allgemeinen Petri-Netzen neu hinzugekommenen Komponenten G und M_0 lassen sich als *Multimengen* charakterisieren (Abschnitt 2.2.2). Es werden nämlich jeweils formalen Objekten natürliche Zahlen zugewiesen. Dabei werden die Multimengen jeweils unterschiedlich interpretiert.

STN=(ST,TR,FR,KP,G,M_0) mit

ST={st_1,st_2,st_3},

TR={tr_1,tr_2},

FR={(st_1,tr_2),(st_2,tr_1),(st_3,tr_1),(tr_1,st_1),(tr_2,st_2),(tr_2,st_3)}

KP(st_m)=∞ für alle $st_m \in$ST

G(st_1,tr_2)=2, G(st_2,tr_1)=1, G(st_3,tr_1)=2, G(tr_1,st_1)=2, G(tr_2,st_2)=1, G(tr_2,st_3)=2,

M_0(st_1)=3, M_0(st_2)=0, M_0(st_3)=0

durch den visuellen Graphen der Abbildung 16 dargestellt werden.

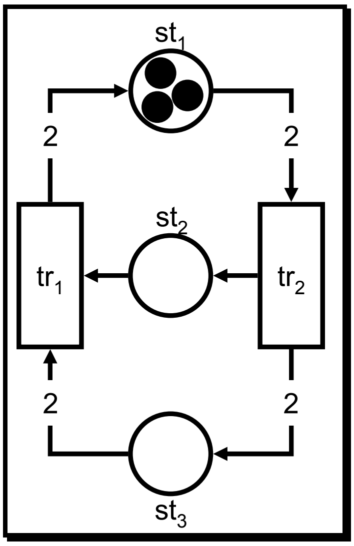

Abbildung 16: Stelle/Transition-Netz STN

3.2.2.2 Struktur von Stelle/Transition-Netzen

3.2.2.2.1 Statische Struktur von Stelle-Transition-Netzen

Die Definition von S/T-Netzen umfasst lediglich statische Aspekte. Es werden nämlich nur jene Bestandteile aufgeführt, die für die Analyse von S/T-Netzen in *einem* Zustand benötigt werden. In den folgenden Abschnitten werden S/T-Netze zu markierten S/T-Netzen mit einer Schaltregel erweitert werden, wodurch auch *zustandsübergreifende* Analysen möglich werden. Hierfür wird zunächst auf die Komponenten eines S/T-Netzes STN sowie ihre formalen und materialen Eigenarten eingegangen.

Durch die *Kapazitätsfunktion* KP wird festgelegt, wie viele Marken[1)] sich in einem Netzzustand auf jeder Stelle höchstens befinden dürfen.[2)] Der Argumentbereich der Kapazitätsfunktion ist die Menge ST aller Stellen. Demnach kann *jeder* Stelle st_m eine *eigene* Kapazität $KP(st_m)$ zugeordnet werden. Der Zielbereich der Kapazitätsfunktion ist die Vereinigung der Menge $\mathbb{N}_+$ mit ∞. Dadurch wird angedeutet, dass auch unendlich große Markenkapazitäten zugelassen sein können, so dass sich auf einer Stelle beliebig viele Marken befinden dürfen. Die Kapazitätsfunktion KP ist als totale Funktion definiert. Demnach muss *jeder* Stelle $st_m \in ST$ im S/T-Netz STN eine Kapazität $KP(st_m)$ zugeordnet werden. Wenn für eine Stelle $st_m \in ST$ nicht *explizit* eine Kapazität $KP(st_m)$ angegeben ist, wird *implizit* ihre unendliche Kapazität $KP(st_m)=\infty$ angenommen. In diesem Fall kann die Stelle beliebig viele Marken aufnehmen.

Die allgemeine[3)] *Gewichtungsfunktion* G, die in für die Definition von S/T-Netzen zugrunde gelegt wurde, ordnet jedem Tupel $(kn_x,kn_y)\in((ST\times TR)\cup(TR\times ST))$ bestehend aus zwei artverschiedenen Knoten $kn_x,kn_y \in KN$ eine natürliche Zahl $G(kn_x,kn_y)\in\mathbb{N}$ zu. Zur Bestimmung der Bilder der Gewichtungsfunktion G wird auf die *spezielle Gewichtungsfunktion*

$$G^*: FR \rightarrow \mathbb{N}_+$$

zurückgegriffen, die in ihrem Argument nur Kanten aus der Flussrelation $FR\subseteq((ST\times TR)\cup(TR\times ST))$ aufnehmen kann. Die Bilder der allgemeinen Gewichtungsfunktion G lassen sich mit Hilfe der speziellen Gewichtungsfunktion G* wie folgt bestimmen:

$$G(kn_x,kn_y)=\begin{cases} G^*(kn_x,kn_y); & \text{falls } (kn_x,kn_y)\in FR \\ 0; & \text{sonst.} \end{cases}$$

Die zugrunde gelegte spezielle Gewichtungsfunktion G* ist eine totale Funktion. Demnach bildet sie jede Kante $(kn_x,kn_y)\in FR$ auf eine positive natürliche Zahl ab. Wenn die

1) Die hier verwendete Darstellung orientiert sich an der üblichen Darstellung in der Literatur. Streng genommen weist die Markierungsfunktion M_z jeder Stelle eine Zahl zu, die angibt, wie viele miteinander identische *Kopien* der *Basismarke* auf einer Stelle st_m liegen. Im Allgemeinen wird der Unterschied zwischen Marken und ihren Kopien aber nicht durchgehalten; für eine Ausnahme vgl. REISIG (1991), S. 152.

2) Die statische Struktur von S/T-Netzen umfasst auch so genannte *Bedingung/Ereignis-Netze* (B/E-Netze). In B/E-Netzen werden lediglich Stellen zugelassen, deren Kapazität maximal 1 sein kann. Entsprechend können die Markierungs- und Gewichtungsfunktionen in ihrer Bildmenge nur die 0 und die 1 aufweisen. Aufgrund dieses bipolaren Charakters der Kapazitätsfunktion werden die Stellen als *Bedingungen* und die Transitionen als *Ereignisse* typisiert. Die Markierung eines B/E-Netzes wird als *Fall* bezeichnet; vgl. Vgl. REISIG (1991), S. 30.

3) Die Bezeichnung „allgemeine" wird im weiteren Verlauf ausgelassen, wenn aus dem Kontext hervorgeht, dass es sich um die allgemeine Gewichtungsfunktion G und nicht um die spezielle Gewichtungsfunktion G* handelt.

spezielle Gewichtung $G^*(kn_x,kn_y)$ einer Kante $(kn_x,kn_y)\in FR$ nicht explizit angegeben ist, wird ihre implizite spezielle Gewichtung $G^*(kn_x,kn_y)=1$ und somit auch $G(kn_x,kn_y)=1$ angenommen.

Die Unterscheidung zwischen der allgemeinen und der speziellen Gewichtungsfunktion G bzw. G* wird nur selten in der Literatur in dieser Form vorgenommen.[1)] Sie vereinfacht jedoch die Formulierung von Integritätsregeln, hinsichtlich derer jedes S/T-Netz STN zulässig sein muss. Insbesondere ist sie für die Formulierung der Integritätsbedingung IB_{STN_1} von Vorteil. Der Integritätsbedingung IB_{STN_1} folgend, gelten alle S/T-Netze als unzulässig, in denen mindestens eine Stelle st_m mit einer Kapazität $KP(st_m)$ vorkommt, wobei Letztgenannte kleiner ist als die Gewichtung mindestens einer der zur Stelle st_m adjazenten Kanten. Wenn die Kapazität $KP(st_m)$ kleiner als die Gewichtung $G(st_m,tr_n)$ einer zur Stelle st_m adjazenten Ausgangskante (st_m,tr_n) ist, müssten durch Schalten von Transition tr_n mehr Marken abgezogen werden als die Stelle st_m aufzunehmen in der Lage ist. Wenn hingegen die Kapazität $KP(st_m)$ kleiner als die Gewichtung $G(tr_n,st_m)$ einer zur Stelle st_m adjazenten Eingangskante ist, müssten durch das Schalten der Transition tr_n auf der Stelle st_m mehr Marken abgelegt werden als die Stelle st_m aufnehmen kann. Darüber hinaus wird mit der Unterscheidung zwischen der allgemeinen und der speziellen Gewichtungsfunktion G bzw. G* die nachbarschaftsbezogene Definition von Schaltvoraussetzungen für Transitionen vereinfacht.[2)]

Wie aus den letzten Erörterungen bereits hervorgeht, ist das Gewicht $G(kn_x,kn_y)$ eines Knotentupels $(kn_x,kn_y)\in FR$ in Abhängigkeit davon zu interpretieren, welcher Knotenart kn_x bzw. kn_y zugehören. Wenn $kn_x\in ST$ gilt, gibt $G(kn_x,kn_y)$ an, wie viele Marken von der Stelle kn_x durch das Schalten der Transition $kn_y\in NB_{ST}(kn_x)$ von der Stelle abgezogen werden müssen. Umgekehrt gibt $G(kn_x,kn_y)$ an, wie viele Marken auf der Stelle $kn_y\in ST$ abgelegt werden müssen, wenn die Transition $kn_x\in VB_{ST}(kn_y)$ schaltet.

3.2.2.2.2 Dynamische Struktur von Stelle/Transition-Netzen

3.2.2.2.2.1 Schaltregeln in S/T-Netzen

3.2.2.2.2.1.1 Schaltregel für einzelne Transitionen in S/T-Netzen

Die dynamische Struktur eine S/T-Netzes STN wird durch die Erweiterung der statischen Struktur (ST,TR,FR,G,W) um eine *Anfangsmarkierungsfunktion* M_0 und eine *Schaltregel* angegeben. Durch die dynamische Struktur wird das potenzielle Verhalten eine S/T-Netzes STN operationalisiert. Sie gibt vor, welche Zustände das STN ausgehend von der Anfangsmarkierungsfunktion M_0 durch die Anwendung der Schaltregel einnehmen kann.

1) Für eine Ausnahme vgl. ZELEWSKI (1995), Bd. 3, S. 33 f.

2) Vgl. Abschnitt 3.2.2.2.2.1.1.

Die *Anfangsmarkierungsfunktion* M_0 bildet jede Stelle st_m auf eine natürliche Zahl $M_0(st_m)\in\mathbb{N}$ ab. Dadurch wird festgelegt, wie viele Marken sich im Initialzustand des S/T-Netzes STN auf jeder Stelle st_m befinden. Dabei gilt für die Ausgangsmarkierung M_0, dass sie entsprechend der Integritätsbedingung IB_{STN_2} einerseits jede Stelle $st_m\in ST$ auf eine nicht-negative natürliche Zahl $M_0(st_m)\geq 0$ abbilden muss und andererseits die stellenspezifische Markierung $M_0(st_m)$ die stellenspezifische Kapazität $KP(st_m)$ nicht übersteigen darf. Aus der Menge aller möglichen Markierungen werden nur solche als *zulässig* bezeichnet, die nicht widersprüchlich zu IB_{STN_2} sind.

Die Familie

$$MF_{STN}=(M_z)_{z\in ZR},\ ZR=\{0,...,Z\} \text{ mit } Z\in\mathbb{N}$$

umfasst alle *möglichen Markierungsfunktionen*[1)]

$$M_z\colon ST \rightarrow \mathbb{N},$$

die für ein S/T-Netz STN denkbar sind. Die Menge ZR aller *Zustände* wird als *Zustandsraum* bezeichnet, wobei der Index $z\in ZR$[2)] einem *Zustand* entspricht, in dem sich das S/T-Netz befindet.[3)] Die Anfangsmarkierungsfunktion M_0 wird entsprechend als *Anfangs-* oder *Initialzustand* des S/T-Netzes angesprochen.

Nicht alle möglichen Markierungen aus der Funktionenfamilie MF_{STN} sind auch zulässig. Die Funktionenfamilie ZMF_{STN} umfasst alle Mitglieder der Familie MF_{STN} aller möglichen Markierungsfunktionen, die hinsichtlich der Integritätsbedingung

$$\widehat{IB_{STN_2}} = \forall M_z\in MF_{STN}, st_m\in ST\colon 0 \leq M_z(st_m) \leq KP(st_m).$$

zulässig sind. Die Integritätsbedingung $\widehat{IB_{STN_2}}$ ist eine Generalisierung der Integritätsbedingung IB_{STN_2}. Während IB_{STN_2} nur auf den Anfangszustand z=0 bezogen ist, bezieht sich die Integritätsbedingung $\widehat{IB_{STN_2}}$ auf alle Zustände $z\in ZR$ mit $ZR=\{0,...,Z\}$ und $Z\in\mathbb{N}$. Insofern ist IB_{STN_2} als Grenzfall in $\widehat{IB_{STN_2}}$ enthalten. Eine *mögliche* Markierung M_z ist genau dann auch *zulässig*, wenn M_z die Integriätsbedingung $\widehat{IB_{STN_2}}$ erfüllt.

Da jede zulässige Markierung auch eine mögliche Markierung ist, muss ZMF_{STN} eine Teilmenge von MF_{STN} sein. Darüber hinaus umfasst die Mengenfamilie $ER_{STN}(M_0)$ alle Markierungen, die durch das Schalten von Transitionen von der Markierung M_0 aus *erreichbar* sind. Wie im weiteren Verlauf aufgezeigt wird, sind alle Markierungen, die durch das Schalten von Transitionen von einer zulässigen Markierung M_0 aus erreichbar sind, auch stets zulässig. Somit ist jede erreichbare Markierung sowohl zulässig als möglich.

1) Für Markierungsfunktionen wird im Folgenden auch die verkürzte Bezeichnung „Markierung“ zugelassen.

2) Der Zustandsindex ist für die Kapazitäts- und Gewichtungsfunktion nicht notwendig, da Kapazitäten und Gewichtungen zustandsinvariante Größen darstellen.

3) Vgl. ZELEWSKI (1995), Bd. 3, S. 33.

Eine Markierungsfunktion M_z ordnet jeder Stelle $st_m \in ST$ im S/T-Netz eine natürliche Zahl als stellenbezogene Markierung $M_z(st_m)$ zu.[1)] Für die Definition von S/T-Netzen wird zunächst nur die Anfangsmarkierungsfunktion M_0 benötigt. Alle Folgemarkierungen, die aus der Anfangsmarkierung M_0 durch die Anwendung einer Schaltregel hervorgehen können, werden im nächsten Abschnitt behandelt. Die Anfangsmarkierungsfunktion M_0 weist jeder Stelle $st_m \in ST$ im S/T-Netz die Anzahl an Marken zu, die im Initialzustand z=0 des S/T-Netzes gültig sind.

Der Anfangszustand eines Petri-Netzes bildet Wissen eines Akteurs bezüglich eines Realitätsausschnitts ab.[2)] Die sonstigen Markierungen M_z, die sich aus dem *Schalten* von Transitionen als Folgemarkierungen von M_0 ergeben, sind hingegen *endogen*. Sie werden nämlich – im Gegensatz zu der Anfangsmarkierung M_0 – nicht „von außen“ vorgegeben, sondern ergeben sich aus der Anwendung einer *Schaltregel*.

Die Schaltregel setzt sich aus zwei Komponenten zusammen. Die erste Komponente der Schaltregel sind die *Schaltvoraussetzungen*. Durch Schaltvoraussetzungen werden zwei Aspekte überprüft, mit denen jeweils sichergestellt wird, dass ein S/T-Netz auch nach dem Schalten einer Transition tr_n bezüglich der Integritätsbedingung $\widehat{IB_{STN_2}}$ zulässig ist.[3)] Eine Unzulässigkeit bezüglich $\widehat{IB_{STN_2}}$ kann nur dann vorliegen, wenn eine Stellenmarkierung $M_z(st_m)$ negativ oder größer als die Kapazität $KP(st_m)$ der betroffenen Stelle st_m ist. Entsprechend wird durch Schaltvoraussetzungen einerseits überprüft, ob für das Schalten einer Transition tr_n in den beanspruchten Stellen genügend Marken vorhanden sind, um das Entstehen einer negativen Stellenmarkierung zu verhindern. Andererseits wird überprüft, ob durch das Schalten einer Transition tr_n für eine Stelle st_m eine Markierung $M_z(st_m)$ bewirkt würde, die größer als die Kapazität $KP(st_m)$ wäre.

Die Schaltvoraussetzung ist der zweiten Komponente der Schaltregel – der *Schaltwirkung* – vorgelagert. Die Untersuchung der Schaltwirkung macht nämlich nur dann Sinn, wenn alle Schaltvoraussetzungen erfüllt sind und somit das Schalten einer Transition tr_n unter einer zulässigen Markierung $M_r(st_m)$ nicht zu einer unzulässigen Markierung $M_f(st_m)$ führen kann. Wenn ihre Schaltvoraussetzungen erfüllt sind und eine Transition tr_n auch unter $M_r(st_m)$ schaltet, muss auch die Markierung $M_f(st_m)$ zulässig sein. Dies wird durch die Schaltfunktion SF_{STN} gewährleistet, mit der bestimmt wird, zu welcher Markierung $M_f(st_m)$ das Schalten einer Transition tr_n unter einer Markierung $M_r(st_m)$ führt.

---

1) Die Zusammenhänge zwischen den einzelnen Elementen der Familie MF_{STN} aller Markierungsfunktionen werden im nächsten Abschnitt weiter vertieft.

2) Vgl. ROSENSTENGEL/WINAND (1991), S. 21, zur Unterteilung in *endogene* und *exogene* Markierungen.

3) Im Gegensatz zu der Integritätsbedingung IB_{STN_1} und IB_{STN_2} ist die Integritätsbedingung $\widehat{IB_{STN_2}}$ für die dynamische Struktur eines S/T-Netzes STN von Bedeutung. In den Integritätsbedingungen IB_{STN_1} und IB_{STN_2} sind nämlich nur zustandinvariante Größen enthalten.

Die Erfülltheit von Schaltvoraussetzungen für eine Transition wird als ihre *Aktiviertheit* bezeichnet. Alternativ wird auch davon gesprochen, die Transition habe *Konzession*.[1)] Die Aktiviertheit einer Transition $tr_n \in TR$ in einem S/T-Netz STN ist unter einer Markierungsfunktion M_z genau dann gegeben, wenn

(1.) $\forall st_m \in VB_{TR}(tr_n)$: $M_z(st_m) \geq G(st_m,tr_n)$,

(2.) $\forall st_m \in NB_{TR}(tr_n) \backslash VB_{TR}(tr_n)$: $M_z(st_m) \leq (KP(st_m) - G(tr_n,st_m))$ und

(3.) $\forall st_m \in NB_{TR}(tr_n) \cap VB_{TR}(tr_n)$: $M_z(st_m) \leq (KP(st_m) + G(st_m,tr_n) - G(tr_n,st_m))$

gelten. Mit der ersten Schaltvoraussetzung wird vermieden, dass nach dem Schalten einer Transition $tr_n \in TR$ eine negative Markierung $M_f(st_m)$ von mindestens einer Stelle st_m im Vorbereich $VB_{TR}(tr_n)$ der Transition bewirkt werden kann. Wie nämlich bei der späteren Präzisierung der Schaltwirkung zu sehen sein wird, werden durch das Schalten einer Transition tr_n von allen Stellen st_m im Vorbereich $VB_{TR}(tr_n)$ der Transition tr_n Marken abgezogen. Die Anzahl der Marken, die abgezogen wird, ist abhängig von der Gewichtung $G(st_m,tr_n)$ der adjazenten Kante (st_m,tr_n), die die Stelle st_m mit der Transition tr_n verbindet. Wenn die Anzahl $G(st_m,tr_n)$ der Marken, die durch das Schalten der Transition tr_n abgezogen werden müssten, größer ist als die Markierung $M_z(st_m)$, dann ist die Schaltvoraussetzung (1.) für tr_n unter M_z nicht erfüllt.

Mit der zweiten Schaltvoraussetzung wird vermieden, dass nach dem Schalten einer Transition tr_n eine Stelle st_m im Nachbereich $NB_{TR}(tr_n)$ von tr_n mit mehr Marken markiert würde als ihr – entsprechend der Kapazitätsfunktion KP – erlaubt ist. Die Anzahl an Marken, die auf den Stellen im Nachbereich der schaltenden Transition tr_n abgelegt werden müssen, richtet sich nach der Gewichtung $G(tr_n,st_m)$ der Kante (tr_n,st_m), die von tr_n zu st_m führt. Wenn die Anzahl $G(tr_n,st_m)+M(st_m)$ der Marken, die nach dem Schalten der Transition tr_n auf einer Stelle st_m, die zwar im Nachbereich $NB_{TR}(tr_n)$, aber nicht im Vorbereich $VB_{TR}(tr_n)$ von tr_n enthalten ist, abgelegt sein müssten, größer ist als die Kapazität $KP(st_m)$ von st_m, dann ist die Schaltvoraussetzung (2.) für tr_n unter M_z nicht erfüllt.

Die dritte Schaltvoraussetzung sorgt dafür, dass keine Transition tr_n aktiviert werden kann, in deren Vor- und Nachbereich $VB_{TR}(tr_n)$ bzw. $NB_{TR}(tr_n)$ eine Stelle st_m vorkommt, deren Markierung $M_z(st_m)$ größer ist als die Kapazität $KP(st_m)$ von st_m zuzüglich der Differenz aus $G(st_m,tr_n)$ und $G(tr_n,st_m)$. Dadurch wird vermieden, dass einerseits von st_m $G(st_m,tr_n)$ Marken abgezogen und andererseits auf st_m $G(tr_n,st_m)$ Marken abgelegt werden und der Nettoeffekt $G(st_m,tr_n)-G(tr_n,st_m)$ dazu führt, dass die Kapazität $KP(st_m)$ überschritten wird.

Entsprechend der früheren Vereinbarung ordnet die allgemeine Gewichtungsfunktion G allen Tupeln $(kn_x,kn_y) \in ((ST \times TR) \cup (TR \times ST))$ bestehend aus zwei artverschiedenen Knoten $kn_x,kn_y \in KN$ eine Gewichtung $G(kn_x,kn_y)$ zu. Als Gewicht von Tupeln (kn_x,kn_y), die nicht in der Flussrelation FR enthalten sind, wurde vereinbart, die Ge-

1) Vgl. STARKE (1990), S. 27.

wichtung $G(kn_x,kn_y)=0$ zu setzen. Demnach ist die Schaltvoraussetzung (2.) ein Unterfall der Schaltvoraussetzung (3.), da in (2.) die Gewichtung $G(st_m,tr_n)=0$ des Tupels $(st_m,tr_n)\notin FR$ ausgelassen wurde.

Mit der ersten Schaltvoraussetzung wird die Priorisierung des Markenkonsums einer schaltenden Transition gegenüber ihrer Markenerzeugung ausgedrückt. In der ersten Schaltvoraussetzung bleibt nämlich unberücksichtigt, ob der Markenverbrauch einer schaltenden Transition tr_n bezüglich einer Stelle $st_m\in(VB_{TR}(tr_n)\cap NB_{TR}(tr_n))$ durch die Markenerzeugung auf st_m so kompensiert werden könnte, dass eine negative Markierung $M_f(st_m)<0$ nicht zustande kommen kann, obwohl $G(st_m,tr_n)>M_r(st_m)$ gilt. Wenn beispielsweise eine Markierung $M_z(st_m)=a$ mit $0\leq a\leq KP(st_m)$ und die beiden Gewichtungen $G(st_m,tr_n)=G(tr_n,st_m)=b$ mit $b>a$ vorliegen, dann ist die Schaltvoraussetzung (1.) für tr_n verletzt, obwohl Markenverbrauch und -erzeugung gleichzeitig stattfinden, so dass die Schaltvoraussetzung (3.) weiterhin erfüllt wird und eine potenzielle Folgemarkierung M_f nicht negativ sein kann. Einerseits müssten nämlich von der Stelle st_m $G(st_m,tr_n)$ Marken abgezogen werden. Andererseits würden auf der Stelle st_m $G(tr_n,st_m)$ Marken abgelegt werden. Beide Effekte würden sich gegenseitig aufheben, so dass die Folgemarkierung M_f mit der Referenzmarkierung M_r übereinstimmt. Ein derartiges Schalten einer Transition tr_n ist allerdings unzulässig, da der Markenkonsum, der für das Schalten von tr_n notwendig wäre, gemäß der Schaltvoraussetzung (1.) nicht zulässig ist.

Die Schaltvoraussetzungen für eine Transition tr_n sind bislang abhängig davon bestimmt, ob die zu tr_n inzidenten Stellen im Vor- oder Nachbereich $VB_{TR}(tr_n)$ bzw. $NB_{TR}(tr_n)$ platziert sind. Für die gesamte Nachbarschaft $NA_{TR}(tr_n)$ der Transition tr_n ausgedrückt, können alle Schaltvoraussetzungen für eine Transition tr_n als

$$\forall st_m\in NA(tr_n)\text{: } G(st_m,tr_n) \leq M(st_m) \leq (KP(st_m) + G(st_m,tr_n) - G(tr_n,st_m))$$

formuliert werden.[1)] In der Formulierung der Schaltvoraussetzungen für die gesamte Nachbarschaft $NA_{TR}(tr_n)$ der Transition tr_n ist die Priorisierung des Markenverbrauchs gegenüber der Markenerzeugung offensichtlicher. Die erste Konjunktionskomponente[2)] $G(st_m,tr_n) \leq M(st_m)$ enthält die Voraussetzung, dass eine Transition nicht mehr Marken verbrauchen darf als auf der Stelle st_m abgelegt sind. Unberücksichtigt bleibt, ob es eventuell eine Kante $(tr_n,st_m)\in FR$ gibt, durch deren Gewichtung $G(tr_n,st_m)$ gleichzeitig auch Marken auf st_m abgelegt würden. Das gleichzeitige Verbrauchen und Konsumieren von Marken wird hingegen bei der zweiten Konjunktionskomponente $(M(st_m) \leq (KP(st_m) + G(st_m,tr_n) - G(tr_n,st_m))$ berücksichtigt. Sie ist für alle Stellen st_m im Nachbereich $NB_{TR}(tr_n)$ von tr_n von Relevanz, umfasst aber auch den Fall, dass st_m im Vorbereich enthalten ist.

1) Vgl. ZELEWSKI (1995), Bd. 3, S. 48.

2) Die Schaltvoraussetzung kann in ihrer entfalteten Variante als Konjunktion

$$\forall st_m\in NA(tr_n)\text{: } (G(st_m,tr_n) \leq M(st_m)) \land (M(st_m) \leq (KP(st_m) + G(st_m,tr_n) - G(tr_n,st_m)))$$

formuliert werden.

Für die Aktiviertheit einer Transition $tr_n \in TR$ in einem S/T-Netz STN unter einer Markierung M_z wird die Aktivierungsrelation AKT_{STN} verwendet.[1] Der Ausdruck $AKT_{STN}(tr_n,M_z)$ gilt genau dann, wenn die o.a. Schaltvoraussetzung erfüllt ist. Somit gilt:

$$\forall tr_n \in TR,\ M_z \in MF_{STN}: AKT_{STN}(tr_n,M_z) \leftrightarrow$$
$$\forall st_m \in NA(tr_n): G(st_m,tr_n) \leq M(st_m) \leq (KP(st_m) + G(st_m,tr_n) - G(tr_n,st_m)).$$

Mit der expliziten Berücksichtigung der gleichzeitigen Markenerzeugung einerseits und des Markenverbrauchs andererseits werden bei den Schaltvoraussetzungen Grenzfälle berücksichtigt, die teilweise in der Literatur unberücksichtigt bleiben. Es handelt sich hierbei in erster Linie um *konstante Schleifen*.[2] Konstante Schleifen sind solche Schleifen, in denen die Gewichtungen $G(st_m,tr_n)$, $G(tr_n,st_m)$ und die Kapazität $KP(st_m)$ übereinstimmen. Diese Konstellation ist in der Abbildung 17 dargestellt.

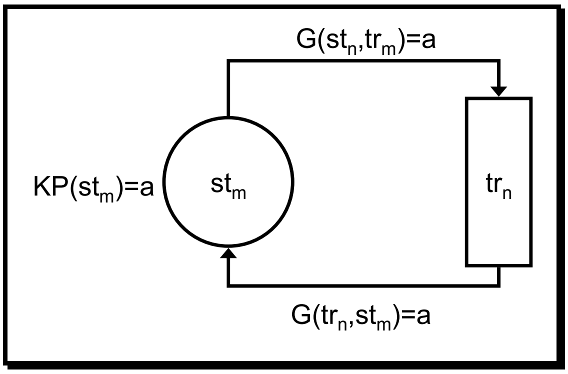

Abbildung 17: Konstante Schleife

Würden bei den Schaltvoraussetzungen der Markenverbrauch von Stellen im Vorbereich $VB_{TR}(tr_n)$ und die Markenerzeugung auf den Stellen im Nachbereich $NB_{TR}(tr_n)$ unabhängig voneinander berücksichtigt, wäre die Aktivierung von Transitionen aus konstanten Schleifen ausgeschlossen. Damit die Transition tr_n aktiviert werden kann, müsste zunächst die Stelle st_m mit $M_z(st_m) \geq a$ markiert sein. Dadurch ist der Markenverbrauch abgesichert. Allerdings würde die alleinige Betrachtung der Markenerzeugung ohne diesen Markenverbrauch dazu führen, dass $G(tr_n,st_m)=a$ Marken auf st_m abgelegt werden müssen, wodurch die Kapazität $KP(st_m)=a$ von st_m überschritten wäre. Aufgrund der wesentlichen Bedeutung, die konstante Schleifen für Ontologie-Netze ha-

1) Vgl. ZELEWSKI (1995), Bd. 3, S. 45.

2) Vgl. ZELEWSKI (1995), Bd. 3, S. 61, mit der Bezeichnung „1-Schleife“.

ben,[1)] wird daher von der vorherrschenden Definition der Schaltvoraussetzungen für Transitionen in S/T-Netzen abgesehen.

Die *Schaltwirkung* hat nur Relevanz für alle Stellen in der Nachbarschaft $NA_{TR}(tr_n)=VB_{TR}(tr_n)\cup NB_{TR}(tr_n)$ der schaltenden Transition tr_n. Die Schaltwirkung umfasst nämlich den Übergang von einer *Referenzmarkierung* M_r zu einer *Folgemarkierung* M_f mit $f,r\in ZR$ und $f=r+1$. Von dem Übergang von M_r zu M_f können nur Stellen im Vor- oder Nachbereich $VB_{TR}(tr_n)$ bzw. $NB_{TR}(tr_n)$ der Transition tr_n betroffen sein. Diese beschränkte Relevanz schaltender Transitionen gilt als grundlegendes Charakteristikum von Petri-Netzen und wird als *Lokalitätsprinzip* bezeichnet.[2)]

Die Schaltwirkung wird durch einen *Schaltakt* ausgelöst. Als Schaltakte werden punktartige Ereignisse bezeichnet, die den Übergang von der einen zu der folgenden Markierung bewirken.[3)] Als ein solches Ereignis kommt in einem Petri-Netz lediglich das Schalten einer Transition in Frage. Außerhalb von Schaltakten sind in Petri-Netzen keine Ereignisse zulässig. Daher ist für den Übergang von einer Markierung zu einer anderen Markierung stets mindestens eine Transition verantwortlich. Nur wenn mindestens eine Transition tr_n geschaltet hat, kommt es zu einem Übergang von einer Referenzmarkierung M_r zu einer Folgemarkierung M_f.

Schaltakte sind Ereignisse, die in punktartiger Form ohne Zeitanspruch stattfinden. Diese Eigenschaft von Schaltakten gilt als ein weiteres grundsätzliches Charakteristikum von Petri-Netzen. In Petri-Netzen ist nämlich grundsätzlich keine Berücksichtigung absoluter Zeitskalen vorgesehen. In Zusammenhang mit zeitlichen Strukturen wird bei Petri-Netzen lediglich von einer *Kausalzeit* gesprochen.[4)] In Petri-Netzen können somit nur zeitliche Anordnungen berücksichtigt werden, wenn Ereignisse in einer kausalen Relation zueinander stehen.[5)] Durch eine derartige kausale Relation kann lediglich eine *relative* zeitliche Anordnung von Zuständen ausgedrückt werden.

Die Schaltregel wird in S/T-Netzen durch die Schaltfunktion

$$SR_{STN}: ZMF_{STN} \times TR \rightarrow ZMF_{STN}$$

---

1) Konstante Schleifen haben in Ontologie-Netzen insbesondere bei der Transformation von Inferenzregeln in *Inferenztransitionen* eine wesentliche Bedeutung. Es handelt sich bei Inferenztransitionen um solche Transitionen tr_n, die grundsätzlich mit den Stellen st_m aus ihrem Vorbereich $VB_{TR}(tr_n)$ durch *Informationskanten* verbunden sind. Informationskanten sind wiederum solche Kanten, die bei einem Schaltakt keine Markenvariation auf ihren adjazenten Stellen bewirken. Auf sämtliche Aspekte wird im Kontext der Transformation von Inferenzregeln in Inferenztransitionen ausführlich eingegangen.

2) Vgl. DESEL/JUHÁS (2001), S. 10; GIRAULT/VALK (2003), S. 11 f.

3) Vgl. ZELEWSKI (1995), Bd. 3, S. 45.

4) Vgl. OBERWEIS (1990), S. 2.

5) Auf diesen bedeutenden Aspekt von Petri-Netzen wird im folgenden Abschnitt näher eingegangen.

operationalisiert.[1)] Die Funktion SR_{STN} bildet jedes Zwei-Tupel (M_r,tr_n), bestehend aus einer zulässigen Referenzmarkierung $M_r \in ZMF_{STN}$ und einer Transition $tr_n \in TR$, auf eine ebenso zulässige Folgemarkierung $SR_{STN}(M_r,tr_n)=M_f$ ab. Dabei sind die Bilder der Schaltfunktion SR_{STN} wie folgt definiert:[2)]

$$\forall st_m \in ST: M_f(st_m) = \begin{cases} M_r(st_m) - G(st_m,tr_n); & \text{falls } st_m \in VB_{TR}(tr_n) \setminus NB_{TR}(tr_n) \\ M_r(st_m) + G(tr_n,st_m); & \text{falls } st_m \in NB_{TR}(tr_n) \setminus VB_{TR}(tr_n) \\ M_r(st_m) - G(st_m,tr_n) + G(tr_n,st_m); & \text{falls } st_m \in NB_{TR}(tr_n) \cap VB_{TR}(tr_n) \\ M_r(st_m); & \text{sonst.} \end{cases}$$

Wenn eine Transition $tr_n \in TR$ unter einer Referenzmarkierung M_r schaltet, wird der Übergang von M_r in die Folgemarkierung $SR_{STN}(M_r,tr_n)=M_f$ bewirkt. Die Markierung M_f ist hierbei von der Markierung M_r und den Gewichtungen $G(kn_x,kn_y)$ der zu tr_n adjazenten Kanten (kn_x,kn_y) abhängig. Der Übergang von der Referenzmarkierung M_r zu der Folgemarkierung M_f durch das Schalten einer Transition tr_n wird auch als

$$M_r[tr_n>M_f$$

angegeben.

Wenn eine Stelle $st_m \in ST$ zwar in Vorbereich $VB_{TR}(tr_n)$, aber nicht im Nachbereich $NB_{TR}(tr_n)$ zu einer Transition tr_n platziert ist, werden von st_m so viele Marken entzogen, wie die Gewichtung $G(st_m,tr_n)$ der adjazenten Kante (st_m,tr_n) angibt. Wenn die Stelle st_m zwar im Nachbereich $NB_{TR}(tr_n)$, aber nicht im Vorbereich $VB_{TR}(tr_n)$ von tr_n enthalten ist, werden auf st_m so viele Marken abgelegt, wie die Gewichtung $G(tr_n,st_m)$ der adjazenten Kante (tr_n,st_m) angibt.

Wenn die Stelle st_m sowohl im Vorbereich $VB_{TR}(tr_n)$ als auch im Nachbereich $NB_{TR}(tr_n)$ von tr_n enthalten ist, tritt ein zweifacher Effekt auf. Zum einen werden von st_m entsprechend der Gewichtung $G(st_m,tr_n)$ der Kante (st_m,tr_n), deren Zielknoten tr_n ist, Marken abgezogen. Zum anderen werden entsprechend der Gewichtung $G(tr_n,st_m)$ der Kante (tr_n,st_m), deren Zielknoten st_m ist, auf st_m Marken abgelegt. Der Gesamteffekt entspricht der Differenz $G(tr_n,st_m)-G(st_m,tr_n)$ zwischen der Anzahl $G(tr_n,st_m)$ aller abzulegenden und der Anzahl $G(st_m,tr_n)$ aller abzuziehenden Marken. Der Gesamteffekt ist positiv, wenn die Gewichtung $G(tr_n,st_m)$ der Kante (tr_n,st_m) größer ist als die Gewichtung

1) Vgl. ZELEWSKI (1995), Bd. 3, S. 44 ff.

2) Die Schaltregel in der hier angegebenen Form weist eine mehrfache Redundanz auf, die allerdings bewusst aus Gründen der natürlichen Schaltregelspezifikation in Kauf genommen wird. Die Schaltregel ließe sich nämlich in kompakterer Form als:

$$M_f(st_m)=M_r(st_m)-G(st_m,tr_n)+G(tr_n,st_m)$$

für alle Stellen $st_m \in ST$ angeben, da die allgemeine Gewichtungsfunktion allen Tupeln (kn_x,kn_y) ein Gewicht von $G(kn_x,kn_y)=0$ zuordnet, wenn $(kn_x,kn_y) \notin FR$ gilt. Die Bilder der beiden Schaltfunktionen sind in jedem Fall gleich.

$G(st_m,tr_n)$ der Kante (tr_n,st_m). Er ist im umgekehrten Fall negativ. Die Markierung $M_r(st_m)=M_f(st_m)$ der Stelle st_m bleibt unverändert, wenn die Gewichtungen $G(tr_n,st_m)$ und $G(st_m,tr_n)$ übereinstimmen.

Die Anzahl der Marken, die durch das Schalten der Transition tr_n von einer Stelle $st_m \in VB_{TR}(tr_n)$ entzogen werden, kann durch eine *transitionsspezifische*[1] *Löschfunktion*

$$tr_n^-: ST \rightarrow \mathbb{N}$$

mit

$$tr_n^-(st_m)=\begin{cases} G(st_m,tr_n); & \text{wenn } st_m \in VB_{TR}(tr_n) \\ 0; & \text{sonst} \end{cases}$$

bestimmt werden.[2] Beim Schalten von tr_n werden demnach *brutto* $G(st_l,tr_l)$ Marken von der Stelle st_l abgezogen.

Die Anzahl der Marken, die durch das Schalten der Transition tr_n auf einer Stelle $st_m \in NB_{TR}(tr_n)$ abgelegt werden, wird wiederum durch eine *transitionsspezifische*[3] *Erzeugungsfunktion*

$$tr_n^+: ST \rightarrow \mathbb{N}$$

mit

$$tr_n^+(st_m)=\begin{cases} G(tr_n,st_m); & \text{wenn } st_m \in NB_{TR}(tr_n) \\ 0; & \text{sonst} \end{cases}$$

bestimmt.[4] Beim Schalten der Transition tr_n werden demnach *brutto* $G(tr_n,st_m)$ Marken auf der Stelle st_l abgelegt.

---

1) Die Löschfunktionen sind insofern transitionsspezfisch, als dass zu jeder Transition $tr_n \in TR$ eine transitionsspezifische Löschfunktion tr_n^- existiert. Die transitionsspezifischen Löschfunktionen $(tr_n^-)_{tr \in TR}$ können somit zu einer Funktionenfamilie trf^- zusammengefasst werden. Ein solches Konstrukt wird allerdings im weiteren Verlauf nicht benötigt.

2) Vgl. STARKE (1990), S. 26; WEBER (2002), S. 40.

3) Analog zu den transitionsspezifischen Löschfunktionen tr_n^- ist jede Erzeugungsfunktion tr_n^+ spezifisch für eine Transition $tr \in TR$ definiert. Auch hierbei können alle transitionsspezifischen Erzeugungsfunktionen $(tr_n^+)_{tr \in TR}$ zu einer Mengenfunktion zusammengefasst werden. Die Familie $trf^+=(tr_n^+)_{tr \in TR}$ hat als Mitglieder alle transitionsspezifischen Erzeugungsfunktionen.

4) Vgl. STARKE (1990), S. 26; WEBER (2002), S. 40.

Die Differenz $(tr_n^+(st_m)\text{-}tr_n^-(st_m))$ entspricht dem *Nettoeffekt* des Schaltens der Transition tr_n. In diesem Nettoeffekt spiegelt sich der „Rollenkonflikt“ von Transitionen aus Schleifen wieder. Er beträgt für konstante Schleifen immer 0, da die Bilder der transitionsspezifischen Erzeugungs- und Löschfunktion tr_n^+ bzw. tr_n^- aufgrund $G(tr_1,st_1)=G(st_1,tr_1)$ übereinstimmen.

Das Schalten einer Transition in dem eingangs durch die Abbildung 16 auf S. 284 illustrierten S/T-Netz wird im Folgenden exemplarisch aufgezeigt:

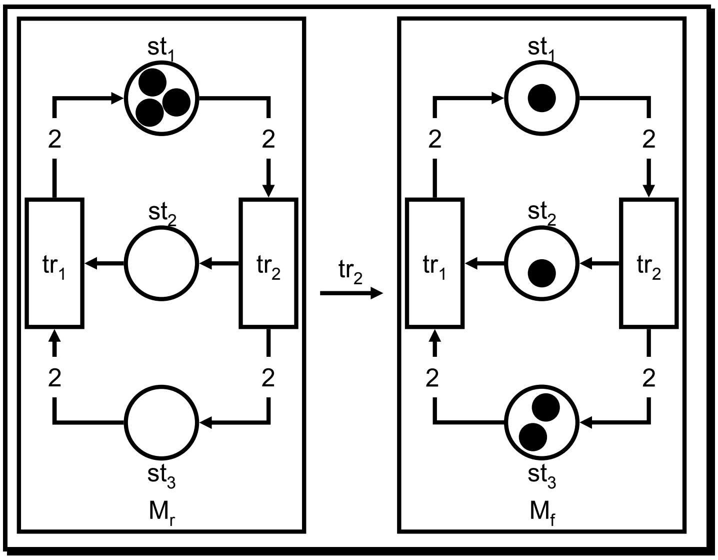

Abbildung 18: Schalten von Transitionen in S/T-Netzen

Der Übergang $M_r[tr_2{>}M_f$ wird in der Abbildung 18 durch den Pfeil zwischen den beiden visuellen Graphen angedeutet. Die Transition tr_2 ist im linken Zustand M_r aktiviert, da die einzige Stelle st_1 im Vorbereich $VB_{TR}(tr_2)$ von tr_2 mit drei Marken markiert ist und die Kante (st_1,tr_2) ein Gewicht $G(st_1,tr_2)$ von 2 hat. Da für die Stellen st_2 und st_3 im Nachbereich $NB(tr_2)$ jeweils unendliche Kapazitäten definiert sind, ist für die Aktivierung von tr_2 irrelevant, dass die Stellen $st_2,st_3 \in NB_{TR}(tr_2)$ im Nachbereich $NB_{TR}(tr_2)$ von tr_2 nicht markiert sind und die Kante (tr_2,st_3) das Gewicht $G(tr_2,st_3)=2$ hat. Somit kann es zum Schaltakt

$$(st_1,3),(st_2,0),(st_3,0)\ [tr_2{>}\ (st_1,1),(st_2,1)(st_3,2)$$

kommen. Die Folgemarkierung M_f ist durch den visuellen Graphen auf der rechten Seite der Abbildung 18 angedeutet.

3.2.2.2.2.1.2 Schaltregel für Transitionsfolgen in S/T-Netzen

Die Schaltwirkung wurde bislang lediglich für den Fall betrachtet, dass eine einzige Transition schaltet. Um Fälle erfassen zu können, in denen mehrere Transitionen nacheinander schalten, werden *Schaltfolgen* eingeführt. Sie dienen der Beschreibung von Übergängen zwischen Markierungen eines Petri-Netzes, die nicht in einem *direkten*

Wirkungszusammenhang stehen müssen. Mit Schaltfolgen können auch *indirekte* Wirkungszusammenhänge zwischen Markierungen beschrieben werden.

Die Menge TR* umfasst alle *Folgen* von Transitionen aus der Menge TR. Die Elemente der Menge TR* werden als *Schaltfolgen* bezeichnet und induktiv wie folgt definiert:

(1.) $\lambda \in TR^*$,

(2.) $\forall tr_n \in TR$: $tr_n \in TR^*$ und

(3.) $\forall sf \in TR^*$, $tr_n \in TR$: $sf\, tr_n \in TR^*$.

Jedes Element $sf \in TR^* \setminus \{\lambda\}$ wird in der Form $sf = tr_1 \ldots tr_{len_{STN}(sf)}$ mit $tr_x \in TR$ für $x = 1, \ldots, len_{STN}(sf)$[1] und $len_{STN}(sf) \in (\mathbb{N} \cup \{\infty\})$ notiert. Die Elemente tr_x müssen nicht paarweise unterschiedlich sein. Eine Transition tr_n kann demnach in einer Schaltfolge $sf \in TR^*$ mehrfach vorkommen. Darüber hinaus umfasst die Menge TR* alle denkbaren Schaltfolgen, somit u.a. auch die Nullschaltfolge λ.

Die Menge TR* aller Schaltfolgen kann hinsichtlich der *Länge* n der Elemente in die zueinander disjunkten Teilmengen

$$TR^* = TR^0 \cup TR^1 \cup \ldots \cup TR^\infty$$

unterteilt werden.[2] Jede Menge $TR^n \subset TR^*$ mit $n \in \mathbb{N}$ umfasst Schaltfolgen der Länge n und wird wie folgt definiert:

(1.) $TR^0 = \{\lambda\}$

(2.) $TR^n = \{sf\, tr_n \mid sf \in TR^{n-1} \wedge tr_n \in TR\}$.

Die Länge n einer Schaltfolge $sf \in TR^n$ wird über die Funktion

$$len_{STN}: TR^* \rightarrow (\mathbb{N} \cup \{\infty\})$$

bestimmt, wobei

(1.) $len_{STN}(\lambda) = 0$ und

(2.) $len_{STN}(tr_n) = 1$ für $tr_n \in TR$

(3.) $len_{STN}(sf\, tr_n) = len_{STN}(sf) + 1$, wenn $sf \in TR^*$ und $tr_n \in TR$

gelten. Für jede längenspezifische Schaltfolgenmenge TR^n gilt demnach:

$$TR^n = \{ sf \mid sf \in TR^* \wedge len_{STN}(sf) = n\}.$$

Der Begriff der Aktivierung von einzelnen Transitionen kann auf Schaltfolgen erweitert werden. Eine Schaltfolge $sf \in TR^*$ mit $sf = (tr_1 \ldots tr_{len_{STN}(sf)})$ ist demnach genau dann unter

1) Die Funktion len_{ST} wird weiter unten als Ausdrucksmittel zur Angabe der Länge einer Schaltfolge vorgestellt.

2) Die lägenspezifische Menge TR^∞ von Schaltfolgen umfasst lediglich solche Schaltfolgen, in denen mindestens eine Transition unendlich oft vorkommt, da S/T-Netze mit einer unendlich großen Menge TR von Transitionen ausgeschlossen wurden. Somit kann keine unendlich große Schaltfolge $sf \in TR^\infty$ existieren, in denen nur paarweise unterschiedliche Transitionen vorkommen.

einer Markierung M_r aktiviert, wenn eine Folge $M_r...M_f$ von Markierungen derart existiert, dass

$$\left(M_r[t_1 > M_{z_1}\right) \wedge ... \wedge M_{z_{len_{STN}(sf)-1}}[tr_{len_{STN}(sf)} > M_f$$

gilt. Um die Aktiviertheit einer Schaltfolge $sf \in TR^*$ mit $sf=(tr_1...tr_{len_{ST}(sf)})$ unter einer Markierung M_r zu repräsentieren, wird der Ausdruck $AKT_{STN}^{SF}(sf,M_r)$ verwendet.

Für den Spezialfall der Nullschaltfolge $\lambda \notin TR$ wird angenommen, dass Sie unter jeder Markierung aktiviert ist:

$$\boxed{\forall M_r \in MF_{STN}: AKT_{STN}^{SF}\left(\lambda,M_r\right)}$$

Für die Definition der Aktiviertheit einer endlichen Schaltfolge $sf \in TR^+$ mit $sf=tr_1...tr_{len_{STN}(sf)}$ wird auf die Aktiviertheit der Transitionen $tr_1,...,tr_{len_{STN}(sf)} \in TR$ unter den jeweiligen Zwischenmarkierungen Bezug genommen. Um die Aktiviertheit der Transitionen auszudrücken, wird wiederum die Relation AKT_{STN} verwendet, die in ihrem Argument eine Transition $tr_n \in TR$ und eine Markierung $M_z \in MF_{STN}$ aufnimmt. Für den Fall einer Schaltfolge $tr_1...tr_{len_{STN}(sf)} \in TR^+$ gilt:

$$\boxed{\begin{gathered}\forall\left(tr_1...tr_{len_{STN}(sf)}\right) \in TR^+,\ M_r \in MF_{STN}: \\ AKT_{STN}^{SF}\left(\left(tr_1...tr_{len_{STN}(sf)}\right),M_r\right) \leftrightarrow \\ \left(AKT_{STN}\left(tr_1,M_r\right) \ \wedge\ SR_{STN}\left(M_r,tr_1\right)=M_{z_1}\right) \wedge ... \wedge \\ \left(AKT_{STN}\left(tr_{len_{STN}(sf)},M_{z_{len_{STN}(sf)-1}}\right) \ \wedge\ SR_{STN}\left(M_{z_{len_{STN}(sf)-1}},tr_{len_{STN}(sf)}\right)=M_f\right).\end{gathered}}$$

Wenn eine Schaltfolge $sf \in TR^*$ mit $sf=tr_1...tr_{len_{STN}(sf)}$ unter einer Markierung M_r aktiviert ist, können die Transitionen $tr_1,...,tr_{len_{STN}(sf)}$ nacheinander geschaltet werden. Nach dem Schalten der letzten Transition $tr_{len_{STN}(sf)}$ wird die Markierung M_f erreicht. Die Markierung M_r, die vor dem sequentiellen Schalten aller Transitionen aus der Schaltfolge sf vorlag, wird als *Startmarkierung* der Schaltfolge sf bezeichnet. Die Markierung M_f, die nach dem sequentiellen Schalten aller Transitionen $tr_1...tr_{len_{STN}(sf)}$ vorliegt, wird als *Schlussmarkierung* der Schaltfolge sf bezeichnet. Die Markierungen

$$M_{z_1},...,M_{z_{len_{STN}(sf)-1}},$$

die zwischen der Startmarkierung M_r und der Schlussmarkierung M_f liegen, werden als *Zwischenmarkierungen* der Schaltfolge sf bezeichnet.

Um den Übergang von einer Startmarkierung M_r zu einer Schlussmarkierung M_f durch das sequentielle Schalten der Transitionen $tr_1,...,tr_{len_{STN}(sf)}$ aus einer Schaltfolge $sf=tr_1...tr_{len_{STN}(sf)}$ formal erfassen zu können, wird die Schaltfunktion SR_{STN} auf Schaltfolgen übertragen. Die partielle Schaltfunktion für Schaltfolgen

$$SR^{SF}_{STN}:\ ZMF_{STN} \times TR^* \rightarrow\ ZMF_{STN}$$

bildet ein Tupel (M_r,sf) – bestehend aus einer Startmarkierung $M_r \in ZMF_{STN}$ und einer zulässigen Schaltfolge $sf \in TR^*$ – auf eine zulässige Schlussmarkierung $M_f \in ZMF_{STN}$ ab, wenn $AKT^{SF}_{STN}(sf,M_r)$ gilt.[1)] Für den Fall einer Schaltfolge sf mit sf=λ gilt:

$$\boxed{\forall M_r \in ZMF_{STN}: SR^{SF}_{STN}\left(M_r,\lambda\right)=M_r}$$

Für denn Fall einer Schaltfolge sf mit $sf \in TR^+$ gilt:

$$\boxed{\begin{array}{c}\forall (tr_1...tr_{len_{ST}(sf)}) \in TR^+;\ M_r, M_{z_1}, ..., M_{z_{len_{STN}(sf)-1}}, M_f \in ZMF_{STN}: \\ SR^{SF}_{STN}\left(M_r,\left(tr_1...tr_{len_{STN}(sf)}\right)\right)=M_f \leftrightarrow \\ \left(SR_{STN}\left(M_r,tr_1\right)=M_{z_1}\right) \wedge ... \wedge \left(SR_{STN}\left(M_{z_{len_{STN}(sf)-1}}, tr_{len_{STN}(sf)}\right)=M_f\right)\end{array}}$$

Somit geht mit dem sequentiellen Schalten aller Transitionen aus einer Schaltfolge $tr_1...tr_n$ mit $SR^{SF}_{STN}\left(\left(tr_1...tr_{len_{STN}(sf)}\right),M_r\right)=M_f$ eine Folge $M_r M_{z_1}...M_{z_{lenSTN(sf)-1}} M_f \in ZMF_{STN}{}^*$ von zulässigen Markierungen einher.

Eine Folge

$$\boxed{\begin{array}{c}\left(M_r,tr_1,M_{z_1}\right)\left(M_{z_1},tr_2,M_{z_2}\right)...\left(M_{z_{len_{STN}(sf)-2}},tr_{n_{len_{STN}(sf)-1}},M_{z_{len_{STN}(sf)-1}}\right)\left(M_{z_{len_{STN}(sf)-1}},tr_{n_{len_{STN}(sf)}},M_f\right) \\ \in \left(ZMF_{STN}\ \times\ TR\ \times\ ZMF_{STN}\right)^* \\ \text{mit } M_r\ \ [tr_1 > M_{z_1} \\ \forall x \in \{2,...,len_{STN}(sf)\text{-}1\}: M_{z_{x-1}}\ \ [tr_x > M_{z_x} \\ \text{und } M_{z_{len_{STN}(sf)-1}}\ \ [tr_{len_{STN}(sf)} > M_f\end{array}}$$

wird als *formaler Prozess* bezeichnet.[2)] Ein formaler Prozess ist demnach – vereinfacht formuliert – eine Folge von Referenzmarkierung-Transition-Folgemarkierung- Tupeln. Jede Folgemarkierung aus einem Tupel stellt die Referenzmarkierung des unmittelbar folgenden Tupels dar. Die erste Referenzmarkierung entspricht der Startmarkierung und die letzte Folgemarkierung der Schlussmarkierung.

Wenn eine Schlussmarkierung M_f aus einer Startmarkierung M_r durch eine endliche Schaltfolge sf hervorgeht, wird die Schlussmarkierung als *erreichbar* aus M_r bezeichnet. Der Zusammenhang wird durch die *schaltfolgenspezifische Erreichbarkeitsrelation*

---

1) Somit handelt es sich bei SR^{SF}_{STN} um eine partielle Funktion, da sie nicht für jedes Tupel (sf,M_r) mit $sf \in TR^*$ und $M_r \in ZMF_{STN}$ definiert ist, sondern nur für solche Tupel, für die $AKT^{SF}_{STN}(sf,M_r)$ gilt.

2) Vgl. ZELEWSKI (1995), Bd. 3, S. 67 f.

$$[sf\!> \subseteq (ZMF_{STN} \times ZMF_{STN})$$

ausgedrückt. Die Elemente aller schaltfolgenspezifischen Erreichbarkeitsrelationen werden in Infix-Notation angegeben. Es wird z.B. M_r [sf> M_f für $(M_r,M_f)\in sf$ notiert, wenn die Schlussmarkierung M_f durch das sequentielle Schalten der Transitionen $tr_1,...,tr_{len_{ST}(sf)}$ mit $sf=tr_1,...,tr_{len_{ST}(sf)}$ von der Startmarkierung M_r aus erreichbar ist.[1)]

Eine Schlussmarkierung M_f ist von einer Startmarkierung M_r aus durch eine endliche Schaltfolge sf erreichbar (M_r [sf> M_f), wenn

(1.) $sf=\lambda$ und somit $M_r=M_f$,

(2.) $sf=tr_n$, $AKT_{STN}(tr_n,M_r)$ und $M_r[tr_n>M_f$ oder

(3.) $sf=tr_1,...,tr_{len_{ST}(sf)}$, $AKT^{SF}_{STN}(sf,M_r)$ und
$\exists M_{z_{lenSTN(sf)-1}}\in ZMF_{STN}$: $(M_r\,[tr_1...tr_{len_{STN}(sf)-1}> M_{z_{lenSTN(sf)-1}}) \wedge (M_{z_{lenSTN(sf)-1}}\,[tr_{len_{STN}(sf)}> M_f)$

gelten.

Im Fall einer Nullschaltfolge λ stimmen Startmarkierung M_r und Schlussmarkierung M_f überein. Die Startmarkierung M_r wird nämlich beibehalten, wenn keine Transition schaltet. Die Schaltfolge $sf=\lambda$ hat in diesem Fall die Länge $len_{STN}(sf)=0$. Darüber hinaus können bei $sf\neq\lambda$ die Start- und die Schlussmarkierung M_r bzw. M_f unterschiedlich sein, müssen es aber nicht.

3.2.2.2.2.2 Nebenläufigkeit und Konflikte

Die Unterstellung kausaler Zeitstrukturen wurde im letzten Abschnitt bereits angesprochen. Aus dieser Annahme werden in diesem Abschnitt zwei weitere Begriffe abgeleitet, die als wesentliche Charakteristika von Petri-Netzen gelten. Es handelt sich dabei um die *Nebenläufigkeit* und *Konflikte*. Sie hängen unmittelbar mit der *kausalen Zeitstruktur* zusammen, die Petri-Netzen zugrunde liegt. Kausale Zeitstrukturen heben sich von *relativen* und *absoluten* Zeitstrukturen[2)] dadurch ab, dass keine *Uhr* vorausgesetzt wird, die Signale aussendet, um eine Zeitmessung vornehmen zu können. Es werden lediglich Interdependenzen zwischen einerseits Systemzuständen und andererseits Ereignissen in der Form konzeptualisiert, dass das potenzielle Stattfinden einer Menge von Ereignissen, das Vorliegen von bestimmten Systemzustände voraussetzt.[3)]

---

1) Diese Schreibweise wurde bereits früher verwendet, als der unmittelbare Übergang von einer Referenzmarkierung M_r zu der Folgemarkierung M_f durch das Schalten einer Transition tr als

$$M_r\,[tr> M_f$$

vorgestellt wurde. Die vorherige Schreibweise stellt den Spezialfall einer schaltfolgenspezifischen Erreichbarkeitsrelation dar, in dem die Schaltfolge sf nur aus einer Transition tr besteht.

2) Zu relativen und absoluten Zeitstrukturen vgl. OBERWEIS (1990), S. 35 ff. u. 39 ff.;

3) Vgl. GENRICH (2002), S. 58; OBERWEIS (1990), S. 25.

Die Begriffe *Nebenläufigkeit* und *Konflikt* knüpfen unmittelbar an den Begriff *Kausalität* an. Während in absoluten Zeitstrukturen beispielsweise die „Parallelität" von Ereignissen erfassbar ist, werden in relativen Zeitstrukturen die gegenseitigen Abhängigkeiten von Ereignissen unabhängig von einer Zeitskala betrachtet. Von Interesse ist lediglich die gegenseitige kausale Abhängigkeit der Aktiviertheiten für die betrachteten Transitionen von Systemzuständen. Dabei stellen Nebenläufigkeit und Konflikt zwei zueinander komplementäre Phänomene dar, die für die Aktivierung von mehreren Transitionen gelten können. Es handelt sich dabei um eine vollkommene Komplementarität, da der Raum aller möglichen Abhängigkeitsbeziehungen zwischen Transitionsaktivierungen durch die nebenläufige und die konfiliktionäre Aktiviertheit vollständig ausgeschöpft wird.

Mit Nebenläufigkeit in Petri-Netzen wird der Umstand bezeichnet, bei dem die Schaltvoraussetzungen für mindestens zwei Transitionen unabhängig voneinander erfüllt sind.[1)] Wenn alle Transitionen aus einer Menge[2)] von nebenläufig unter einer Referenzmarkierung M_r aktivierten Transitionen gemeinsam schalten, tritt eine Folgemarkierung M_f ein, die auch dann eintreten muss, wenn alle Transitionen aus dieser Menge in beliebiger Reihenfolge nacheinander schalten würden. Die Folgemarkierung

$$M_f = SR_{STN}^{SF}\left(\left(tr_1 ... tr_n\right), M_r\right)$$

würde somit aus dem Schalten jeder Schaltfolge sf, die einer Permutation aller Transitionen $tr_1,...,tr_n$ aus der o.a. Menge entspricht, hervorgehen.

Das gemeinsame Schalten nebenläufig aktivierter Transitionen führt immer zu einer zulässigen Folgemarkierung M_f, wobei die Zulässigkeit von M_f genau dann gegeben ist, wenn sie nicht im Widerspruch zu der Integritätsbedingung $\widehat{IB_{STN_2}}$ ist. Bemerkenswert hierbei ist, dass eine Transition tr_n auch nebenläufig „zu sich selbst" aktiviert sein kann. Dieser Sonderfall tritt genau dann ein, wenn tr_n unter einer solchen Referenzmarkierung M_r aktiviert ist, der zufolge einerseits alle Stellen $st_m \in VB_{TR}(tr_n)$ jeweils solche stellenbezogenen Markierungen aufweisen, dass der mehrfache Markenkonsum durch tr_n ge-

---

1) Vgl. GIRAULT/VALK (2003), S. 12.

2) Bei einer formalsprachlichen Präzisierung der nebenläufigen oder konfliktionären Aktivierung von Transitionen müsste eine solche Menge von Transitionen als *Multi*menge aufgefasst werden, da sich die nebenläufige bzw. konfliktionäre Aktiviertheit auch auf die „multiple Aktiviertheit" von *einer* Transition beziehen kann; vgl. BAUMGARTEN (1996), S. 97. Solche multiplen Aktiviertheiten werden weiter unten thematisiert. Der Verfasser verzichtet allerdings aus zwei Gründen auf eine formalsprachlich aufwendige Präzisierung von Nebenläufigkeit und Konflikt. Ersten liegen bereits Ansätze vor, in denen nebenläufige und konfliktionäre Aktiviertheiten formal formuliert werden; vgl. ZELEWSKI (1995), Bd. 3, S. 74 ff., allerdings lediglich mit binären Relationen, mittels derer die entweder nebenläufige oder konfliktionäre Aktiviertheit von genau zwei Transitionen ausgedrückt werden kann. Zweitens wird der Aufwand, der mit der formalen Präzisierung unweigerlich verbunden ist, im Kontext der Modellierung kooperativer Informationssysteme durch keinen entsprechenden Bedarf gerechtfertigt.

währleistet ist und andererseits alle Stellen $st_m \in NB_{TR}(tr_n)$ solche stellenbezogenen Markierungen $M_r(st_m)$ und Kapazitäten $KP(st_m)$ aufweisen, dass es durch das mehrfache Schalten von tr_n zu keiner Kapazitätsüberschreitung kommt. Für die folgenden Ausführungen werden jedoch lediglich die „konventionellen“ Fälle der entweder nebenläufigen oder konfliktionären Aktivierung von genau zwei voneinander unterschiedlichen Transitionen untersucht. Die Untersuchung kann allerdings nicht auf die Aktiviertheit von mindestens drei Transitionen erweitert werden. Zu beachten ist hierbei, dass drei Transitionen nicht notwendigerweise gemeinsam nebenläufig aktiviert sein müssen, wenn die betrachteten Transitionen paarweise nebenläufig aktiviert sind.[1)] Es kann nämlich durchaus sein, dass die betrachteten Transitionen jeweils zu zweit gemeinsam ohne Integritätsverletzung schalten können und somit nebenläufig aktiviert sind, jedoch das gemeinsame Schalten aller Transitionen zu einer Integritätsverletzung führen würde.

Beispielhaft ist die nebenläufige Aktiviertheit von Transitionen in der Abbildung 19 illustriert:

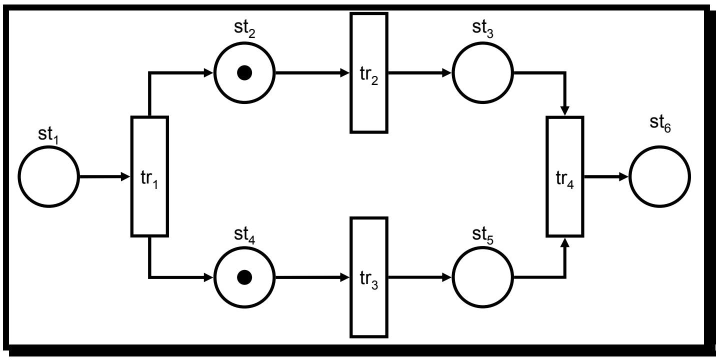

Abbildung 19: Nebenläufigkeit in Petri-Netzen

Im S/T-Netz der Abbildung 19 sind die beiden Transitionen tr_2 und tr_3 nebenläufig zueinander aktiviert. Durch die Schaltfolgen $sf_1=tr_2tr_3$ und $sf_2=tr_3tr_2$ würde – ausgehend von der Referenzmarkierung M_r entsprechend Abbildung 19 – stets die gleiche Folgemarkierung M_f mit den stellenbezogenen Markierungen

$$M_f(st_1)=M_f(st_2)=M_f(st_4)=M_f(st_6)=0$$
$$\text{und } M_f(st_3)=M_f(st_5)=1$$

hervorgebracht.

Nebenläufigkeit tritt im offensichtlichsten Fall dann auf, wenn zwei Transitionen tr_1 und tr_2 unter derselben Markierung aktiviert sind und in ihren jeweiligen Nachbarschaften $NA_{TR}(tr_1)$ bzw. $NA_{TR}(tr_2)$ keine gemeinsame Stelle st_m aufweisen. In diesem Fall können die Transitionen tr_1 und tr_2 in beliebiger Reihenfolge oder „gleichzeitig“ schalten. Die Folgemarkierungen sind stets die gleichen.

1) Vgl. BAUMGARTEN (1996), S. 94 f.

Der nicht so offensichtliche Fall tritt dann auf, wenn die nebenläufige Aktiviertheit von Transitionen gegeben ist, in deren Nachbarschaften $NA_{TR}(tr_1)$ und $NA_{TR}(tr_2)$ mindestens eine Stelle st_m gemeinsam vorkommt.

Wenn die Stelle st_m in den Vorbereichen $VB_{TR}(tr_1)$ und $VB_{TR}(tr_2)$ vorkommt, sind bei nebenläufiger Aktiviertheit auf st_m genügend Marken enthalten, um beide Transitionen schalten zu lassen. Die Markierung $M_r(st_m)$ der Stelle $st_m \in (VB_{TR}(tr_1) \cap VB_{TR}(tr_2))$ ist in diesem Fall mindestens so groß wie die Summe $G(st_m,tr_1)+G(st_m,tr_2)$ der Gewichte der adjazenten Kanten (st_m,tr_1) und (st_m,tr_2).

Wenn die Stelle st_m in den Nachbereichen $NB_{TR}(tr_1)$ und $NB_{TR}(tr_2)$ der beiden Transitionen tr_1 und tr_2, aber in keiner der beiden Mengen $VB_{TR}(tr_1)$ und $VB_{TR}(tr_2)$ vorkommt[1)], muss die Kapazität $KP(st_m)$ der Stelle st_m bei nebenläufiger Aktiviertheit mindestens so groß sein wie die Summe aus ihrer aktuellen Markierung $M_r(st_m)$, der Gewichtung $G(tr_1,st_m)$ und der Gewichtung $G(tr_2,st_m)$. Durch das Schalten beider Transitionen tr_1 und tr_2 würden auf der Stelle insgesamt so viele Marken abgelegt werden, dass – trotz ihrer aktuellen Markierung – ihre Kapazität nicht überschritten wird.

Wenn eine Stelle im Nachbereich $NB_{TR}(tr_1)$ und nicht im Vorbereich $VB_{TR}(tr_1)$ der ersten Transition tr_1 sowie im Vorbereich $VB_{TR}(tr_2)$ und nicht im Nachbereich $NB_{TR}(tr_2)$ der zweiten Transition tr_2 enthalten ist, liegt eine sequentielle Anordnung der Transitionen vor. In diesem Fall muss die Stelle st_m, die zwischen tr_1 und tr_2 liegt, eine Kapazität $KP(st_m)$ haben, die mindestens so groß ist wie die Summe $M_r(st_m)+G(tr_1,st_m)$ ihrer Markierung $M_r(st_m)$ und der Anzahl $G(tr_1,st_m)$ an Marken, die durch das Schalten von tr_1 auf st_m abgelegt werden. Gleichzeitig muss die Markierung $M_r(st_m)$ mindestens so groß sein wie die Anzahl $G(st_m,tr_2)$ an Marken, die durch das Schalten der Transition tr_2 der Stelle st_m entzogen werden.

Der zur Nebenläufigkeit komplementäre Fall der Aktiviertheit zweier unterschiedlicher Transitionen umfasst die *konfliktionäre Aktiviertheit*. Der Zustand einer konfliktionären Aktiviertheit zweier unterschiedlicher Transitionen wird als *Konflikt* bezeichnet. Aufgrund der Exhaustion der Aktivierungsmöglichkeiten für mindestens zwei Transitionen liegt ein Konflikt zwischen mindestens zwei aktivierten Transitionen immer dann vor, wenn die Transitionen nicht nebenläufig aktiviert sind. Unabhängig von der Nebenläufigkeit kann ein Konflikt zwischen zwei aktivierten Transitionen als ein Zustand definiert werden, bei dem das gleichzeitige Schalten von mehreren Transitionen, die jede für sich aktiviert sind, zu einer *unzulässigen* Markierung führen würde. Die Unzulässigkeit einer Markierung M_z liegt in S/T-Netzen genau dann vor, wenn M_z im Widerspruch zu der Integritätsbedingung $\widehat{IB_{STN_2}}$ steht. Ein solcher Konflikt ist in der Abbildung 20 illustriert.

1) Kommt die Stelle st_m sowohl in $NB_{TR}(tr_1)$ und $NB_{TR}(tr_2)$ als auch in mindestens einer der beiden Mengen $VB_{TR}(tr_1)$ oder $VB_{TR}(tr_2)$ vor, liegt mindestens eine Schleife vor. Dieser Fall wird weiter unten diskutiert.

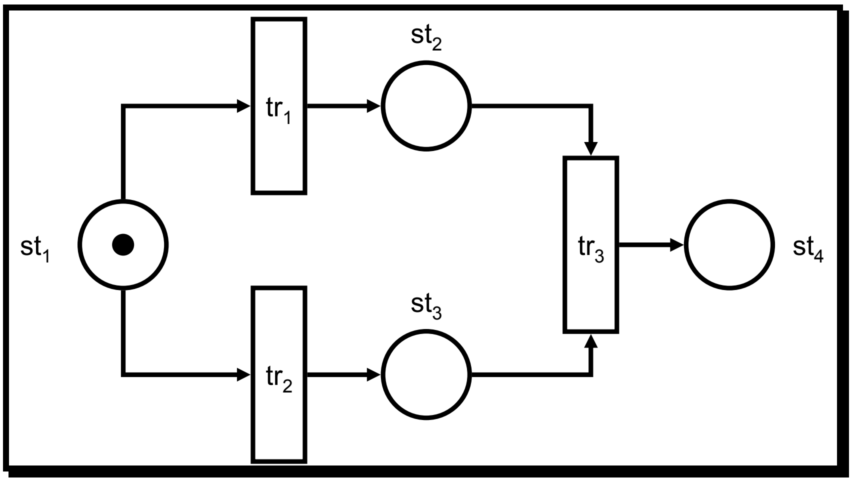

Abbildung 20: Konflikt im allgemeinen Fall

In der Abbildung 20 sind die Transitionen tr_1 und tr_2 konfliktionär aktiviert. Wenn nämlich beide Transitionen gemeinsam schalten würden, müssten der Stelle st_1 insgesamt zwei Marken entzogen werden. Aufgrund ihrer aktuellen Markierung mit nur einer Marke würde dies allerdings zu einer negativen und somit unzulässigen Markierung führen.

Im Fall konfliktionär aktivierter Transitionen kann das Schalten der einen Transition die Aktiviertheit der anderen Transition aufheben. Wenn das Schalten der einen Transition die Aktiviertheit der anderen Transition aufhebt, liegt grundsätzlich ein Konflikt zwischen den beiden Transitionen vor. Das Aufheben der Aktiviertheit der einen Transition durch das Schalten der zweiten Transition ist zwar *hinreichend*, allerdings nicht *notwendig* für das Vorliegen eines Konflikts.[1] Wenn nämlich eine Stelle in zwei Schleifen enthalten ist, können die an den Schleifen beteiligten Transitionen zueinander in Konflikt stehen, obwohl durch das Schalten der einen Transition die Aktiviertheit der anderen Transition nicht aufgehoben wird. Der Umstand wird in Abbildung 21 verdeutlicht.

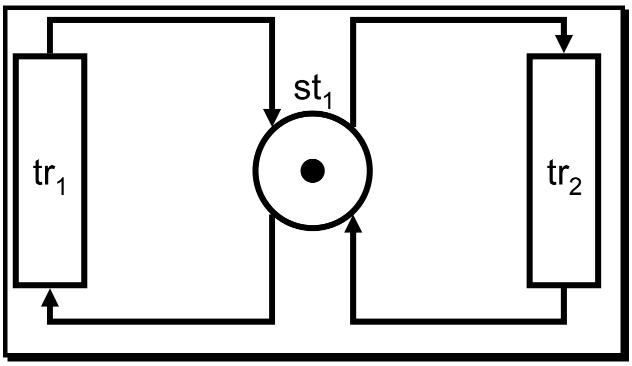

Abbildung 21: Konflikt bei zwei Schleifen

1) Vgl. ZELEWSKI (1995), Bd. 3, S. 76.

Die beiden Transitionen tr_1 und tr_2 sind konfliktionär aktiviert, da ihr gleichzeitiges Schalten eine negative Markierung auf der Stelle st_1 bedeuten würde. Durch das Schalten einer Transition wird allerdings nicht die Aktiviertheit der jeweils anderen Transition beeinflusst.

Umgekehrt kann allerdings aus dem Umstand, dass beim Schalten einer Transition tr_1 die Aktiviertheit der zweiten Transition nicht aufgehoben wird, nicht auf Nebenläufigkeit geschlossen werden. Die Nicht-Aufhebung der Aktiviertheit einer zweiten Transition ist zwar *notwendig*, aber nicht *hinreichend* für die nebenläufige Aktiviertheit.[1)] Für die Bestimmung der Nebenläufigkeit muss nämlich das punktartige gemeinsame Schalten zweier Transitionen betrachtet werden. Nur wenn das gemeinsame Schalten von zwei Transitionen zu einer zulässigen Markierung führen würde, sind die Transitionen nebenläufig aktiviert. Es sind wiederum Schleifen mit einer gemeinsamen Stelle, in denen ein Konflikt – und somit keine Nebenläufigkeit – vorliegen kann. Beispielsweise wird durch das Schalten eine Transition in Abbildung 21 die Aktiviertheit der zweiten Transition nicht beeinflusst. Dennoch sind die Transitionen nicht nebenläufig aktiviert.

1) Vgl. ZELEWSKI (1995), Bd. 3, S. 77.

4 Integration von Ontologien und Petri-Netzen

4.1 Überblick über das Integrationsvorhaben

Durch den Übergang von S/T-Netzen auf *höhere* Petri-Netze wird die Petri-Netz-Theorie um die Möglichkeit angereichert, *individualisierte* Marken zu verwenden. Durch diesen Übergang werden die Marken – die bislang in S/T-Netzen lediglich atomare formale Objekte darstellten – mit einer *inneren Struktur* versehen. Die innere Struktur jeder Marke trägt dazu bei, dass alle Marken individuell und somit voneinander unterscheidbar sein können. Mit dieser Individualisierung von Marken ist in erster Linie der Vorteil verbunden, Petri-Netz-gestützte Modelle kompakter darzustellen.[1)] Während in S/T-Netzen für jedes individuell zu repräsentierende Objekt eine eigene Stelle konzipiert werden muss, können in höheren Petri-Netzen unterschiedliche Objekte auf derselben Stelle platziert werden. Die Netz-Dynamik wird dann in Abhängigkeit von den Objekteigenschaften formuliert.

In den folgenden Abschnitten werden die beiden Bausteine des integrativen Modellierungsansatzes zusammengeführt. Es handelt sich hierbei einerseits um Ontologien und andererseits um höhere Petri-Netze. Durch die Integration von Ontologien und Petri-Netzen wird zum einen erhofft, die fehlende operationale Komponente von Ontologien zu überbrücken. Zum anderen wird eine Petri-Netz-Klasse entwickelt, die eine sprachadäquate Modellierung[2)] durch ihre Ontologie-Komponente erlaubt. Dieser Zusammenhang wird durch Tabelle 11 verdeutlicht.

	sprachadäquate Modellierung	algebraisch-prädikatenlogische Basis	operationale Semantik
ONTOLOGIEN	✓	✓	
HÖHERE PETRI-NETZE		✓	✓

Tabelle 11: Integrationspotenzial von Ontologien und höheren Petri-Netzen

Mit S/T-Netzen wurde im vorherigen Abschnitt eine Grundform *elementarer* Petri-Netze vorgestellt. Mit dem Übergang zu *höheren* Petri-Netzen wird das Petri-Netz-Konzept um mehrere Facetten angereichert. Dabei ist die Bezeichnung „höhere Petri-

1) Vgl. GENRICH (1987), S. 208 ff.; ZELEWSKI (1995), Bd. 5.1, S. 8 ff.

2) Die Sprachadäquatheit der ontologiegestützten Modellierung wird durch ihre syntaktischen, semantischen und pragmatischen Aspekte gewährleistet. Sie wurde in den Abschnitten 3.1.2, 3.1.3 bzw. 3.1.4 vorgestellt.

Netze" teilweise mit unterschiedlichen Verständnissen belegt.[1] Von den verschiedenen Begriffsverständnissen ist dasjenige am weitesten verbreitet, bei dem solche Petri-Netze als „höhere" Petri-Netze bezeichnet werden, deren Marken grundsätzlich voneinander unterscheidbar sind. In weiteren Konzeptualisierungen werden höhere Petri-Netze mit hierarchisierten Netzstrukturen oder Erweiterungen um temporale Aspekte assoziiert. Um alternative Begriffsverständnisse für die vorliegende Arbeit nicht gänzlich auszuschließen, werden sie unter dem Begriff der „höheren Petri-Netze i.w.S." subsumiert. Als „höhere Petri-Netz i.e.S." werden hingegen Petri-Netze der erstgenannten Art bezeichnet. Für die vorliegende Arbeit wird von diesem letztgenannten Begriffsverständnis ausgegangen. Demnach handelt es sich bei höheren Petri-Netzen i.e.S. um solche Petri-Netze, deren Marken individueller Natur sein können. Falls nicht explizit hervorgehoben, wird im weiteren Argumentationsverlauf von höheren Petri-Netzen i.e.S. ausgegangen, wenn der Zusatz „i.e.S." entfällt.

S/T-Netze erweisen sich als äußerst unpraktisch, wenn komplexe Modelle konstruiert werden sollen. Für jedes individuell zu repräsentierende Objekt, das im Modell berücksichtigt werden soll, muss in S/T-Netzen eine eigene Stelle vorgesehen werden. Es können zwar Objekte derselben Art zusammengefasst und durch Markierung einer Stelle repräsentiert werden, allerdings kann auch diese Vorgehensweise zu einer drastischen Zunahme der Netzkomplexität führen.

In höheren Petri-Netzen werden ähnliche Objekte zusammengefasst und durch die Markierung einer Stelle repräsentiert. Dadurch können kompaktere Modelle konstruiert werden, die äquivalent zu Modellen mit einer höheren Komplexität sind, die mit Hilfe von elementaren Petri-Netzen gestaltet werden. Ausgehend von dem Vorhaben, durch individuelle Marken auch solche Petri-Netz-gestützten Modelle mit einer geringen Komplexität der Netztopologie konstruieren zu können, die für praktische Problemfelder relevant sein können, wurden verschiedene Klassen höherer Petri-Netze entwickelt[2]. Grundlegend sind dabei *algebraische Petri-Netze*, *farbige Petri-Netze* und *Pr/T-Netze*. In algebraischen Petri-Netzen[3] werden die Stellen eines Petri-Netzes mit Sorten einer algebraischen[1] sortierten Spezifikation, die Flussrelationen mit

1) Zu Überblicken über unterschiedliche Auslegungen der Bezeichnung *höhere Petri-Netze* vgl. GEROGIANNIS ET AL. (1998), S. 139 ff.; VAN DER AALST/VAN HEE (2002), S. 41 ff.

2) Für einen umfassenden Vergleich der verschiedenen Klassen höherer Petri-Netze vgl. GEROGIANNIS ET AL. (1998), S. 139 ff. Für eine generische Beschreibung des Prinzips höherer Petri-Netze vgl. SMITH (1998), S. 181.

3) Vgl. REISIG (1991A), S. 1 ff.

einer algebraischen[1] sortierten Spezifikation, die Flussrelationen mit Multimengen von Termen und die Transitionen mit Gleichungen über der Signatur annotiert. In farbigen Petri-Netzen[2] wird jede Stelle im Petri-Netz mit einer „Farbe“, die in der Regel einem Datentyp aus einer zugrunde gelegten höheren Programmiersprache entspricht, annotiert und die Transitionen mit Formeln annotiert, die nur für die Datentypen aus den Annotationen der adjazenten Stellen definiert sind. In Pr/T-Netzen[3] werden die Netzkomponenten mit Ausdrücken über einer prädikatenlogischen Signatur beschriftet.

Ontologie-Netze weisen sowohl Aspekte von Pr/T-Netzen als auch von algebraischen Netzen auf. Für beide Petri-Netz-Klassen wird die Prädikatenlogik als ein formaler Baustein des Integrationskonzepts herangezogen. Sie wurden als besonders geeignet für das Integrationskonzept beurteilt, da für Ontologien bereits ein formaler Rahmen gewählt wurde, der zur Spezifikation mit sortierten Signaturen „abwärtskompatibel“ ist. Dieser Rahmen kann reibungslos in eine sortierte Prädikatenlogik überführt werden, die als Grundlage für Pr/T-Netze verwendet werden kann. Darüber hinaus verhalten sich Pr/T-Netze sowohl zu algebraischen als auch zu farbigen Petri-Netzen insofern komplementär, als dass keine Unterschiede in der Ausdruckmächtigkeit ausgemacht werden können.

1) Als *algebraische sortierte Signaturen* werden solche Signaturen bezeichnet, in denen nur Sorten und Operationssymbole spezifiziert werden können. Wird zudem ein ausgezeichnetes Relationssymbol (z.B. „=“) zugelassen, das in der zugehörigen logischen Struktur als Gleichheitsrelation interpretiert wird, wird das Kalkül, das die Signatur und die Formierungsregeln für algebraische Formeln umfasst, als *algebraische Gleichungslogik* bezeichnet. Eine algebraische Signatur in Verbindung mit einer Menge von Formeln, die alle mit Hilfe des Relationssymbols für die Gleichheit konstruiert werden, wird als *algebraische Spezifikation* bezeichnet.

2) Vgl. JENSEN (1996), S. 65 ff.

3) Vgl. HE ET AL. (2004), S. 11 ff.; KORCZYNSKI ET AL. (1990), S. 40 ff.

4.2 Ontologie-Netze

4.2.1 Definition von Ontologie-Netzen

Ein Ontologie-Netz ON ist definiert als:

$$ON=(ST,TR,FR,IR,SPEZ_{OS},trans,A_{SIG_{ST}},AN_{ON},KP,M_0)$$

mit

IB_{PN_1}: $(ST \cap TR) = \varnothing$,
IB_{PN_2}: $(ST \cup TR) \neq \varnothing$,
IB_{PN_3}: $(ST \cup TR) = (VB_{FR}(FR) \cup NB_{FR}(FR))$,
$IB_{ON_{1.1}}$: $\forall (st_m,tr_n)\in(ST\times TR), w\in K^+$: $typ_{RS_{OS}}(AN_{ST}(st_m)) = w \rightarrow AN_{FR}(st_m,tr_n)\in MULT(TT_w)$,
$IB_{ON_{1.2}}$: $\forall (tr_n,st_m)\in(TR\times ST), w\in K^+$: $typ_{RS_{OS}}(AN_{ST}(st_m)) = w \rightarrow AN_{FR}(tr_n,st_m)\in MULT(TT_w)$,

IB_{ON_2}: $\forall (st_m,tr_n)\in(ST\times TR), w\in K^+$: $typ_{RS_{OS}}(AN_{ST}(st_m)) = w \rightarrow AN_{IR}(st_m,tr_n)\in MULT(TT_w)$,
IB_{ON_3}: $\forall st_m\in ST, w\in K^+$: $typ_{RS_{OS}}(AN_{ST}(st_m)) = w \rightarrow M_0(st_m)\in MULT(GTT_w)$

IB_{ON_4}: $\forall st_m\in ST$: $0 \leq \#(M_0(st_m)) \leq KP(st_m)$ und

IB_{ON_5}: $\forall x\in VAR, tr_n\in TR$: $x\in var_{OUT}(tr_n) \rightarrow (x\in var_{IN}(tr_n) \vee x\in var_{F_{OS}}(AN_{TR}(tr_n)))$.

Die Komponenten eines Ontologie-Netzes sind:

(1.) eine Menge $ST=\{st_1,...,st_M\}$ von *Stellen* mit $m\in\{1,...,M\}$ und $M\in\mathbb{N}_+$ mit $ST=ST_{POS} \cup ST_{NEG}$,

(2.) eine Menge $TR=\{tr_1,...,tr_N\}$ von *Transitionen* mit $n\in\{1,...,N\}$ und $N\in\mathbb{N}_+$ mit $TR = TR_{PROZ} \cup TR_{DEK}$ mit $TR_{PROZ} \cap TR_{DEK}=\varnothing$ und $TR_{DEK} = TR_{INF} \cup TR_{INT}$ mit $TR_{INT} \cap TR_{INF} = \varnothing$ mit
$\forall tr_n\in TR_{DEK}: \neg\exists st_m\in ST: (st_m,tr_n)\in FR$ und
$\forall tr_n\in TR_{INF}: |(NB(tr_n)| = 1$,
$\forall tr_n\in TR_{INT}: \exists st_m\in ST: (st_m,tr_n)\in IR$,
$\forall tr_n\in TR_{INT}: NB_{TR}(tr_n)=\varnothing$,

(3.) eine *Flussrelation* FR mit $FR\subseteq((ST\times TR)\cup(TR\times ST))$,

(4.) eine *Informationsrelation* $IR\subseteq(ST\times TR_{DEK})$,

(5.) eine *Ontologie* $SPEZ_{OS}=(SIG_{OS},INF_{SIG_{OS}},INT_{SIG_{OS}})$ mit einer zugrunde liegenden *ontologischen Signatur* SIG_{OS} mit
$SIG_{OS}=(K,MEN,ALPH_{META},\sqsubseteq,\triangleq,\curlyvee,OPS,typ_{OPS_{OS}},RS,typ_{RS_{OS}},VARF_{SIG_{OS}},bezf,deff)$ mit
$SIG_{DY}=(K,MEN,ALPH_{META},\sqsubseteq,\triangleq,\curlyvee,OPS,typ_{OPS_{OS}},RS_{DY},typ_{RS_{DY}},VARF_{SIG_{OS}},bezf,deff)$,
$SIG_{ST}=(K,MEN,ALPH_{META},\sqsubseteq,\triangleq,\curlyvee,OPS,typ_{OPS_{OS}},RS_{ST},typ_{RS_{ST}},VARF_{SIG_{OS}},bezf,deff)$ und
$RS = RS_{ST} \cup RS_{DY}$,

einer Menge $INF_{SIG_{OS}} \subseteq FORM_{SIG_{OS}}$ von *Inferenzregeln* über SIG_{OS} und
einer Menge $INT_{SIG_{OS}} \subseteq FORM_{SIG_{OS}}$ von *Integritätsregeln* über SIG_{OS},

(6.) eine Familie $trans=(trans_{INT},trans_{INT})$ von *Zuordnungsfunktionen* für deklarative Transitionen mit:
$trans_{INF}$: $INF_{SIG_{OS}} \rightarrow TR_{INF}$ und
$trans_{INT}$: $INT_{SIG_{OS}} \rightarrow TR_{INT}$,

(7.) einem *Support* $A_{SIG_{ST}}$
mit $A_{SIG_{ST}}=(OBF_{OS},OPF,RF_{ST},IF_{ST})$ und
$\forall F\in(INF_{SIG_{OS}}\cup INT_{SIG_{OS}})$: $F\in FORM_{SIG_{ST}} \rightarrow A_{SIG_{ST}} \models F$,

(8.) eine Familie $AN_{ON}=(AN_{ST},AN_{TR},AN_{FR},AN_{IR})$ von *Annotationsfunktionen* mit

(8.1.) der *Stellenannotation*
AN_{ST}: $ST \rightarrow RS_{DY}\cup\underline{RS}_{DY}$,
mit $\underline{RS}_{DY}=\{\underline{R}_j \mid R_j\in RS_{DY}\}$.

(8.2.) der *Transitionsannotation*
AN_{TR}: $TR \rightarrow FORM_{SIG_{ST}}$,

(8.3.) der *operationalen Kantenannotation*
AN_{FR}: $(ST\times TR)\cup(TR\times ST) \rightarrow MULT(TT_{SIG_{OS}})$,

(8.4.) der *deklarativen Kantenannotation*
AN_{IR}: $(ST\times TR) \rightarrow MULT(TT_{SIG_{OS}})$,

(9.) eine Kapazitätsfunktion KP: $ST \rightarrow \mathbb{N}_{+}\cup\{\infty\}$ und

(10.) eine Anfangsmarkierungsfunktion M_0: $ST \rightarrow MULT(GTT_{SIG_{OS}})$.

Bezüglich ihrer Netztopologie[1)] unterscheiden sich Ontologie-Netze von S/T-Netzen in einigen Punkten. Erstens umfasst zwar auch jedes Ontologie-Netz ON eine Menge ST von *Stellen*, eine Menge TR von *Transitionen* und eine *Flussrelation* $FR\subseteq((ST\times TR)\cup(TR\times ST))$[2)], allerdings werden die Stellen und Transitionen nochmals

1) Die Topologie eines Ontologie-Netzes ist die Gesamtheit aus der Knotenmenge ST, der Transitionenmenge TR, der Flussrelation FR mit $FR\subseteq((ST\times TR)\cup(TR\times ST))$, der Informationsrelation $IR\subseteq(ST\times TR_{DEK})$ sowie den Integritätsbedingungen $IB_{ON_{1.1}}$, $IB_{ON_{1.2}}$, IB_{ON_2}, IB_{ON_3} und IB_{ON_4}.

2) Eigentlich werden in Ontologie-Netzen nur Flusskanten von Stellen zu prozeduralen Transitionen und von Transitionen zu Stellen zugelassen. Flusskanten sind nämlich bei einem Schaltakt stets mit einer Markenvariation verbunden. Deklarative Transitionen dürfen hingegen keine Markenvariation auf ihren Vorbereichsstellen bewirken. Somit werden Flusskanten von Stellen zu deklarativen Transitionen ausgeschlossen. Entsprechend müsste die Flussrelation FR definiert werden als:

$$FR\subseteq((ST\times TR_{PROZ})\cup(TR\times ST)).$$

nach ihren *Arten* unterschieden. So werden einerseits Stellen in *positive* und *negative* Stellen und andererseits Transitionen in *prozedurale* und *deklarative* Transitionen unterschieden. Zweitens sind in jedem Ontologie-Netz ON noch weitere topologische Komponenten enthalten, die für S/T-Netze nicht vorgesehen sind. Bezüglicher dieser topologischen Unterschiede zwischen S/T-Netzen einerseits und Ontologie-Netzen andererseits ist insbesondere die Informationsrelation IR hervorzuheben, auf die in Kürze eingegangen wird.

Für die Netztopologie eines Ontologie-Netzes ON gelten die Integritätsbedingungen, IB_{PN_1}, IB_{PN_2} und IB_{PN_3}, die für alle Petri-Netze Gültigkeit haben.[1] Einerseits gilt für alle Ontologie-Netze die Disjunktheitsbedingung $ST \cap TR = \varnothing$ (IB_{PN_1}). Das heißt, dass es in einem Ontologie-Netz ON kein formales Objekt geben darf, das sowohl eine Stelle als auch eine Transition ist. Andererseits gilt für alle Ontologie-Netze die Existenzbedingung $ST \cup TR \neq \varnothing$ (IB_{PN_2}). Zudem gilt in Ontologie-Netzen die Verknüpftheitsbedingung IB_{PN_3}, derzufolge kein isolierter Knoten vorkommen darf. Erst durch die Verknüpftheitsbedingung IB_{PN_3} wird gewährleistet, dass die variablen Extensionen von dynamischen Relationssymbolen netzendogen durch Schaltakte variiert werden können.[2] Aufgrund der Existenzbedingung IB_{PN_2} und der Verknüpftheitsbedingung IB_{PN_3} sind die kleinstmöglichen Ontologie-Netzen wohldefiniert. Es handelt sich hierbei um die Ontologie-Netze $ON_1=(\{st_m\},\{tr_n\},\{(st_m,tr_n)\})$ und $ON_2=(\{st_m\},\{tr_n\},\{(tr_n,st_m)\})$.[3]

Stellen aus Ontologie-Netzen können in *positive* und *negative Stellen* unterschieden werden. Positive Stellen sind solche Stellen, die durch die Stellenannotationsfunktion

---

Hierdurch würde auch die für Petri-Netze charakteristische Bipartitheit nicht durchbrochen werden, da die Menge TR_{PROZ} eine Teilmenge der Menge TR aller Transitionen ist. Allerdings würde hierdurch die Übertragbarkeit der Erkenntnisse aus S/T-Netzen auf Ontologie-Netze beeinträchtigt werden. Bei Bedarf kann nämlich die Netztopologie eines Ontologie-Netzes auf jene Aspekte reduziert werden, die auch für S/T-Netze gültig sind. Zu der Netztopologie eines S/T-Netzes gehören neben den Integritätsbedingungen $IB_{PN_{1.1}}$, $IB_{PN_{1.2}}$, IB_{PN_2} und IB_{PN_3}, die Knotenmenge ST, die Transitionenmenge TR und die Flussrelation FR mit $FR \subseteq ((ST \times TR) \cup (TR \times ST))$.

1) Vgl. Abschnitt 3.2.1.

2) Da Informationskanten auf den schaltbedingten Markenab- oder -zufluss von bzw. zu einer adjazenten Stelle st_m keinen Einfluss haben, brauchen sie in der Verknüpftheitsbedingung IB_{PN_3} nicht berücksichtigt zu werden. Durch die Verknüpftheitsbedingung IB_{PN_3} wird gewährleistet, dass die Extensionen der dynamischen Relationssymbole, die den Stellen in einem Ontologie-Netz zugeordnet sind, durch Schaltakte variiert werden können. Da Informationskanten jedoch weder für den Markenabfluss noch für den Markenzufluss verwertbar sind, werden sie in der Verknüpftheitsbedingung IB_{PN_3} nicht berücksichtigt. Wenn die einzige Verknüpfung einer Stelle st_m zu den restlichen Netzkomponenten durch eine Informationskante $(st_m,tr_n) \in IR$ erlaubt würde, so gäbe es keine Möglichkeit, durch Schaltakte auf die Extension des Relationssymbols R_j mit $AN_{ST}(st_m)=R_j$ einzuwirken. Die Abfragen der nicht variierbaren Extensionen von Relationssymbolen in der Transitionsannotation $AN_{TR}(tr_n)$ erfolgen über die statische SIG_{ST}-Struktur $A_{SIG_{ST}}$.

3) Vgl. hierzu die Ausführungen zu den kleinstmöglichen allgemeinen Petri-Netzen PN_1 und PN_2 in Abschnitt 3.2.1.

AN_{ST} einem (positiven) Relationssymbol R_j aus der zugrunde gelegten Ontologie $SPEZ_{OS}$ zugeordnet sind. Die Menge ST_{POS} umfasst alle positiven Stellen aus einem Ontologie-Netz ON. Negative Stellen sind hingegen solche Stellen, die mit einem negativen Relationssymbol $\underline{R}_j$ annotiert werden. Jedes negative Relationssymbol $\underline{R}_j$ entspricht der Negation eines positiven Relationssymbols R_j.[1)] Die Menge ST_{NEG} umfasst alle negativen Stellen aus einem Ontologie-Netz ON. Mit den Markierungen von positiven Stellen werden mittelbar[2)] die zustandsspezifischen Extensionen der positiven Relationssymbole angegeben, die den positiven Stellen zugeordnet sind. Analog dazu werden mit den Markierungen der negativen Stellen mittelbar die zustandsspezifischen Extensionen der negativen Relationssymbole angegeben, die den negativen Stellen zugeordnet sind.

Die Menge TR aller Transitionen aus einem Ontologie-Netz ON wird in die Mengen TR_{PROZ} aller *prozeduralen* Transitionen und die Menge TR_{DEK} aller *deklarativen* Transitionen unterteilt.[3)] Die Menge TR_{DEK} aller deklarativen Transitionen wird zusätzlich in die Menge TR_{INF} aller *Inferenztransitionen* und die Menge TR_{INT} aller *Integritätstransitionen* unterteilt. Sowohl prozedurale als auch deklarative Transitionen bewahren die grundsätzliche Eigenschaft von Transitionen aus Petri-Netzen, *dynamische* Knoten zu sein. Ausprägungen beider Transitionsarten werden nämlich dazu genutzt, Übergänge zwischen unterschiedlichen Markierungen einzuleiten.[4)] Der Unterschied zwischen prozeduralen und deklarativen Transitionen liegt hingegen in der Art der Aktivität, die durch sie repräsentiert werden.

---

1) Vgl. Abschnitt 3.1.4.3.3.2.

2) Die Extension eines dynamischen Relationssymbols $R_j \in RS_{DY}$ geht insofern *mittelbar* aus der Markierung der entsprechenden Stelle hervor, als dass es sich bei der Stellenmarkierung stets um ein Multimenge über der Menge $GTT_{SIG_{OS}}$ aller ontologischen *Grundterm*tupel handelt. Im Unterschied zu anderen höheren Petri-Netz-Klassen (vgl. z.B. STARKE (1990), S. 240) werden die Stellen aus Ontologie-Netzen nicht mit formalen Objekten aus einer SIG_{OS}-Struktur $A_{SIG_{OS}}$ markiert, sondern mit Termen. Erst aus der – eindeutig festgelegten – extensionalen Interpretation der Marken gehen die Extensionen dynamischer Relationssymbole hervor. Die extensionale Interpretation der Marken aus Ontologie-Netzen wird weiter unten diskutiert.

3) Der Unterscheidung zwischen prozeduralen und deklarativen Transitionen entspricht im Umfeld aktiver Datenbanksysteme die Unterscheidung zwischen *Event-Condition-Action-Regeln* und *deduktiven Regeln*; vgl. DITTRICH/GATZIU (2000), S. 35 f.

4) Diese Aussage gilt für Integritätstransitionen lediglich eingeschränkt. Im Unterschied zu prozeduralen Transitionen und Inferenztransitionen werden nämlich Integritätstransitionen mit der Absicht spezifiziert, unzulässige Markierungen kenntlich zu machen. Eine Markierung M_z ist genau dann bezüglich einer Integritätstransition $tr_n \in TR_{INT}$ unzulässig, wenn es eine Substitution θ gibt, unter der tr_n in M_z aktiviert ist. Um die Unzulässigkeit der Markierung M_z bezüglich einer Integritätstransition tr_n aufzuheben, bedarf es in der Regel eines integritätsherstellenden Eingriffs in das Ontologie-Netz „von außen“. Demnach ist in Bezug auf Integritätstransitionen lediglich ihre Aktiviertheit von Interesse, aber weniger die Folgemarkierung, die durch ihr unzulässiges Schalten hervorgerufen würde.

Prozedurale Transitionen haben ihren Ursprung in *originär* operationalen Eingriffen in die Netzmarkierung. Entsprechend wird durch das Schalten einer prozeduralen Transition $tr_n \in TR_{PROZ}$ der Übergang von einer Referenzmarkierung M_r zu einer Folgemarkierung[1)] M_f repräsentiert, der einem Ereignis aus einem faktischen oder gedachten Realitätsausschnitt entspricht.

Deklarative Transitionen entsprechen hingegen *derivativen* operationalen Eingriffen in die Netzmarkierung. Dabei gehen die derivativ operationalen Eingriffe aus deklarativen Inferenz- oder Integritätsregeln hervor.[2)] Zwar wird auch durch das Schalten von deklarativen Transitionen der Übergang von Referenzmarkierungen zu Folgemarkierungen repräsentiert,[3)] jedoch entspricht das Schalten einer deklarativen Transition – wie noch später aufgezeigt wird – stets einem *Wissenszuwachs* des Akteurs und keinem Ereignis aus dem faktischen oder gedachten Realitätsausschnitt. Durch das Schalten einer Inferenztransition wird aus dem Wissen des Akteurs bezüglich der Gültigkeit von Formeln „neues" Wissen bezüglich der Gültigkeit anderer Formeln erschlossen. Der Erschließung neuen Wissens durch das Schalten von Inferenztransitionen entspricht eine Erweiterung der Markierung. Eine Erweiterung der Markierung entspricht der Erzeugung neuer Marken bei gleich bleibender sonstiger Markierung. Eine Verkleinerung entspricht hingegen dem Abzug von Marken bei gleich bleibender sonstiger Markierung. Eine derartige Verkleinerung der Markierung durch Inferenztransitionen kommt nicht in Frage. Eine Verkleinerung der Markierung würde einen Abzug von Marken von Stellen voraussetzen, der für deklarative Transitionen grundsätzlich ausgeschlossen ist. Deklarative Transitionen – und somit auch Inferenztransitionen – können mit ihren Vorbereichsstellen lediglich durch Informationskanten verbunden sein, über die keine Marken abgezogen werden können.

Für Integritätstransitionen sind hingegen solche Schaltakte nicht Betrachtungsgegenstand. Bereits die Aktiviertheit von Integritätstransitionen ist Indikator für eine unzulässige Netzmarkierung.

Durch das Schalten einer prozeduralen Transition muss es nicht notwendig zu einer Erweiterung der Markierung kommen. Die Möglichkeit eines Markenabzugs wird durch das Schalten von prozeduralen Transitionen genauso zugelassen wie der Fall einer Markenablage.

---

1) Die Begriffe *Referenz-* und *Folgemarkierung* werden analog zu ihrer Verwendung bei S/T-Netzen verwendet. Sie werden im Kontext der dynamischen Struktur von Ontologie-Netzen präzisiert.

2) Der Koexistenz von deklarativen und prozeduralen Transitionen in Ontologie-Netzen entspricht im Datenbankenbereich die Koexistenz von *deduktiven* und *aktiven* Regeln; vgl. LAUSEN ET AL. (1998), S. 71 ff. Deduktive Regeln stimmen mit den hier thematisierten Inferenzregeln überein. Aktive Regeln werden dazu verwendet, zustandsändernde Grundoperationen in Datenbanken durchzuführen.

3) Für Integritätstransitionen gilt diese Annahme unter dem Vorbehalt, dass ihre Aktiviertheit einen unzulässigen Kognitionszustand des Modellierungsträgers repräsentiert. Ausgehend von solchen Zuständen interessieren Schaltakte von Integritätstransitionen nicht weiter.

Die Zuordnung von Inferenz- und Integritätstransitionen zu Inferenz- bzw. Integritätsregeln erfolgt durch die beiden Funktionen aus (6.) der Definition von Ontologie-Netzen:

$$\text{trans}_{\text{INF}}\text{: } \text{INF}_{\text{SIG}_{\text{OS}}} \rightarrow \text{TR}_{\text{INF}}$$
$$\text{und trans}_{\text{INT}}\text{: } \text{INT}_{\text{SIG}_{\text{OS}}} \rightarrow \text{TR}_{\text{INT}}.$$

Die linkstotale und bijektive Funktion $\text{trans}_{\text{INF}}$ ordnet jeder Inferenzregel $F \in \text{INF}_{\text{SIG}_{\text{OS}}}$ aus der Ontologie SPEZ_{OS} genau eine Inferenztransition $\text{trans}_{\text{INF}}(F)=tr_n$ mit $tr_n \in \text{TR}_{\text{INF}}$ zu. Sie ist einerseits linkstotal, da jeder Inferenzregel $F \in \text{INF}_{\text{SIG}_{\text{OS}}}$ aus der zugrunde gelegten Ontologie SPEZ_{OS} eine Inferenztransition $\text{trans}_{\text{INF}}(F)=tr_n$ im Ontologie-Netz ON entsprechen muss. Andererseits ist sie bijektiv, da jede *Inferenztransition* $tr_n \in \text{TR}_{\text{INF}}$ *genau einer* Inferenzregel $F \in \text{INF}_{\text{SIG}_{\text{OS}}}$ entsprechen muss (vice versa).

Die Funktion $\text{trans}_{\text{INT}}$ ordnet jeder Integritätsregel $F \in \text{INT}_{\text{SIG}_{\text{OS}}}$ aus der Ontologie SPEZ_{OS} genau eine Integritätstransition $\text{trans}_{\text{INT}}(F)=tr_n$ mit $tr_n \in \text{TR}_{\text{INT}}$ zu. Im Gegensatz zur Funktion $\text{trans}_{\text{INF}}$ ist die Funktion $\text{trans}_{\text{INT}}$ allerdings nicht rechtstotal. In der Menge TR_{INT} können nämlich auch solche Integritätstransitionen enthalten sein, die keinen Integritätsregeln aus der zugrunde gelegten Ontologie SPEZ_{OS} zugeordnet sind. Es kann sich hierbei beispielsweise um solche Integritätstransitionen handeln, die benötigt werden, um den gegenseitigen Ausschluss von Marken zu positiven und negativen Stellen zu bewahren.

In der Informationsrelation IR sind nur Kanten enthalten, die stets eine Stelle st_m als Ursprungs- und eine deklarative[1)] Transition tr_n als Zielknoten haben. Sie werden – zwecks Unterscheidung von Flusskanten – als *Informationskanten* bezeichnet.[2)] Bei Informationskanten handelt es sich um Tupel $(st_m,tr_n) \in (\text{ST} \times \text{TR}_{\text{DEK}})$, die nicht für Markenab- oder -zuflüsse vorgesehen sind. Solche Markenab- und -zuflüsse können in allen Petri-Netz-Klassen nur aus Schaltakten der adjazenten Transitionen resultieren. Marken*zuflüsse* über Informationskanten kommen grundsätzlich nicht in Frage, da sie stets als Eingangsknoten eine Stelle st_m und als Ausgangsknoten eine deklarative Transition $tr_n \in \text{TR}_{\text{DEK}}$ aufweisen. Der Zufluss von Marken zu einer Stelle st_m kann allerdings nur über Eingangstransitionen von st_m erfolgen. Darüber hinaus sind Informationskanten auch nicht für den Marken*abfluss* verwertbar. Informationskanten werden lediglich dazu verwendet, die Marken aus ihren adjazenten Stellen zu „lesen".[3)] Daher unterscheiden sich Informationskanten aus der Informationsrelation IR wesentlich von den Flusskanten der Flussrelation FR. Die Annotation von Informationskanten ist zwar für

1) Somit werden Informationskanten mit adjazenten prozeduralen Transitionen für Ontologie-Netze ausgeschlossen. Die Verknüpfung mit Stellen durch Informationskanten bleibt dadurch den deklarativen Transitionen vorbehalten.

2) Alternativ finden sich oftmals auch die Bezeichnungen *Lesekante* (vgl. CHRISTENSEN/HANSEN (1993), S. 186; BALDAN ET AL. (2000), S. 442; JAESCHKE (1996), S. 37; RÖBBECKE (1995), S. 59) oder *Testkante* (vgl. WIENBERG (2001), S. 141).

3) Vgl. BALDAN ET AL. (2000), S. 443.

die Schaltvoraussetzung ihrer adjazenten Transitionen von Bedeutung, aber nicht für deren Schaltwirkung.

Informationskanten haben für Ontologie-Netze u.a. aufgrund zweier Aspekte eine hohe Bedeutung. Der erste Aspekt betrifft die *formale* Charakterisierung von Informationskanten. Wie bereits erwähnt, werden Informationskanten dazu verwendet, die zustandsspezifischen Markierungen von Stellen zwar zu lesen, aber nicht zu verändern. Diese Charakteristik von Informationskanten kommt der Transformation von Inferenz- und Integritätsregeln zu Inferenz- bzw. Integritätstransitionen zu Gute. Im Fall von Inferenztransitionen wird durch Informationskanten überprüft, ob die Bedingungen erfüllt sind, die hinreichend für das Schalten einer Inferenztransition sind. Damit mehrere Inferenztransitionen nebenläufig zueinander aktiviert werden können, die mindestens eine gemeinsame Stelle in ihren Vorbereichen aufweisen, müssen die entsprechenden Stellen über Informationskanten zu den Transitionen verbunden sein. Ansonsten könnten Inferenztransitionen u.U.[1)] nicht nebenläufig schalten.[2)]

Der zweite Aspekt von Informationskanten betrifft ihre *materiale* Charakterisierung. Diese materiale Sichtweise knüpft unmittelbar an die nebenläufige Aktivierbarkeit von Transitionen an, die adjazent zu Informationskanten sind. Eine solche nebenläufige Aktivierbarkeit von Transitionen hat im Rahmen der Modellierung kooperativer Informationssysteme dann eine hohe Bedeutung, wenn mehreren Akteuren zwar die Einsicht von Ressourcen, aber nicht deren Manipulation erlaubt werden soll. Diese materiale Sichtweise kommt allerdings nur bedingt zur Geltung, da Informationskanten in Ontologie-Netzen nur Stellen und deklarative Transitionen miteinander verbinden können. Informationskanten mit adjazenten prozeduralen Transitionen werden hingegen nicht zugelassen.[3)]

---

1) Die weiteren Umstände, die dazu führen können, dass Inferenztransitionen konfliktionär zueinander aktiviert sind, werden weiter unten thematisiert.

2) Eine solche Konzeptualisierung findet sich beispielsweise bei DAHR (1994), S. 19 ff., und RADA ET AL. (1990), S. 58, im Rahmen der Transformation von Regeln aus logischen Programmen. Die dort vorzufindenden Transitionen können nicht nebenläufig schalten. Sie müssen nacheinander geschaltet werden, damit sie als Inferenzen vollzogen werden können.

3) Die Beschränktheit von Informationskanten auf adjazente deklarative Transitionen kann nicht streng begründet werden, sondern entspricht einer konzeptionellen Basisentscheidung des Verfassers. Die Entscheidung lässt sich lediglich anhand einiger Plausibilitätsargumente rechtfertigen. Zum einen wird mit dem Ausschluss von prozeduralen Transitionen als adjazente Knoten zu Informationskanten die Separation zwischen deklarativen und prozeduralen Netzkomponenten teilweise aufrechterhalten. Denn Informationskanten lassen sich dadurch gemeinsam mit Inferenz- und Integritätstransitionen als deklarative Netzkomponenten klassifizieren. Durchbrochen wird diese Separation allerdings durch Flusskanten, die als Eingangsknoten deklarative Transitionen und als Ausgangsknoten Stellen aufweisen. Als Eingangsknoten kommen hierbei in erster Linie Inferenztransitionen in Betracht, durch deren Schalten Fakten expliziert werden, die zuvor implizit in der Markierung des Netzes enthalten waren. Insofern lässt sich die Flussrelation FR in einem Ontologie-Netz ON sowohl als deklarative als auch als prozedurale Netzkomponente charakterisieren.

Der *Support* eines Ontologie-Netzes ON ist eine SIG_{ST}-Struktur $A_{SIG_{ST}}$. Mit dem Support $A_{SIG_{ST}}$ wird der statische Teil der Ontologie $SPEZ_{OS}$, die dem Ontologie-Netz zugrunde liegt, extensional interpretiert. Das heißt, dass von der ontologischen Signatur SIG_{OS}, auf der die Ontologie $SPEZ_{OS}$ basiert, zwar alle Konzepte und Operationssymbole aus SIG_{OS}, allerdings nur die statischen Relationssymbole aus RS_{ST} extensional interpretiert werden. Die Extensionen der dynamischen Relationssymbole aus RS_{DY} gehen hingegen mittelbar aus der zustandsbezogenen Markierung von ON hervor. Durch die Gesamtheit der extensionalen Interpretationen der statischen Relationssymbole in der statischen SIG_{ST}-Struktur $A_{SIG_{ST}}$ und der extensionalen Interpretation dynamischer Relationssymbole durch die zustandsbezogenen Markierungen von ON werden alle Relationssymbole aus SIG_{OS} extensional interpretiert. Dabei ist zu beachten, dass der Support $A_{SIG_{ST}}$ des Ontologie-Netzes ON auch dann aufgeführt werden muss, wenn es sich bei allen Relationssymbolen aus SIG_{OS} um dynamische Relationssymbole handelt, deren Extensionen aus der zustandsbezogenen Markierung hervorgehen. Denn der Support $A_{SIG_{ST}}$ umfasst auch die zustandsinvarianten Mengenfamilien OBF_{OS} und OPF. Insbesondere die Familie OBF_{OS} aller konzeptspezifischen Objektmengen wird benötigt, um die stellenbezogenen Markierungen zu Extensionen von dynamischen Relationssymbolen auswerten zu können. Denn Ontologie-Netze setzen grundsätzlich eine Objektmenge OB voraus, die aus der Vereinigung $OB_{k_1} \cup ... \cup OB_{k_n}$ aller konzeptspezifischen Objektmengen aus OBF_{OS} hervorgeht. Da es sich bei den stellenbezogenen Markierungen stets um Multimengen von ontologischen Grundtermtupeln handelt, wird zudem die Mengenfamilie OPF benötigt, um die Grundterme, die vornehmlich Konstantensymbolen entsprechen, zu null-stelligen Operationen – so genannten Konstanten – aus OPF und somit zu Individuen aus OB auswerten zu können.

Der Support $A_{SIG_{ST}}$ eines Ontologie-Netzes ON muss eine wesentliche Anforderung erfüllen. Es müssen alle Inferenz- und Integritätsregeln aus der zugrunde gelegten Ontologie $SPEZ_{OS}$, die ausschließlich mit statischen Relationssymbolen konstruiert sind, in $A_{SIG_{ST}}$ gültig sein. Im Fall von Inferenzregeln wird dadurch verhindert, dass für ein On-

Zum anderen werden mit dem Ausschluss von prozeduralen Transitionen als adjazente Knoten zu Informationskanten *asymmetrische Konflikte* vermieden, die dann entstehen können, wenn die gleiche Stelle st_m mittels einer eingehenden Informationskante (tr_1,st_m) von einer Transition tr_1 und mittels einer Flusskante (st_m,tr_2) zu einer Transition tr_2 verbunden ist. Ein asymmetrischer Konflikt zwischen tr_1 und tr_2 liegt genau dann vor, wenn sowohl tr_1 als auch tr_2 aktiviert sind und das Schalten von tr_1 zwar keinen Einfluss auf die Aktiviertheit von tr_2 hätte, allerdings das Schalten von tr_2 die Aktiviertheit von tr_1 aufheben würde (zu asymmetrischen Konflikten vgl. BALDAN ET AL. (2000), S. 443; CIARDO/ZIAL (1996), S. 280 f.). Dadurch, dass in Ontologie-Netzen für Informationskanten nur deklarative Transitionen als Ausgangsknoten erlaubt werden und deklarativen Transitionen gegenüber prozeduralen Transitionen eine höhere *Aktivierungspriorität* eingeräumt wird, können asymmetrische Konflikte in dieser Form vermieden werden.

Um das „Lesen“ von Marken durch prozedurale Transitionen zu erlauben, kann auf die Konstruktion von *Schleifen* zurückgegriffen werden. Durch die beiden Flusskanten $(st_m,tr_n),(tr_n,st_m) \in FR$ mit gleicher Flusskantenannotation können durch das Schalten der prozeduralen Transition tr_n von der Stelle st_m Marken abgezogen und gleichzeitig wieder auf dieser Stelle st_m abgelegt werden.

tologie-Netz ON mit zugrunde liegender Ontologie $SPEZ_{OS}$ ein Support verwendet wird, in dem nicht das gesamte Wissen erschlossen ist, das bezüglich einer „statischen“ Inferenzregel in dem Support enthalten sein müsste. Im Fall von „statischen“ Integritätsregeln wird durch ihre Gültigkeit in $A_{SIG_{ST}}$ vermieden, dass ein Support verwendet wird, in dem auch Formeln, die bezüglich Integritätsregeln unzulässig sind, gültig sind.

Bei der Darstellung von Ontologie-Netzen durch visuelle Graphen werden weiterhin zur Repräsentation von Stellen runde Symbole und für die Repräsentation von Transitionen Rechtecke verwendet. Kanten werden zudem weiterhin durch Pfeile dargestellt. Um die formale und materiale Unterscheidung zwischen unterschiedlichen Stellen-, Transitionen- und Kantenarten auch in der graphischen Visualisierung berücksichtigen zu können, werden die Symbole entsprechend der Abbildung 22 verwendet:

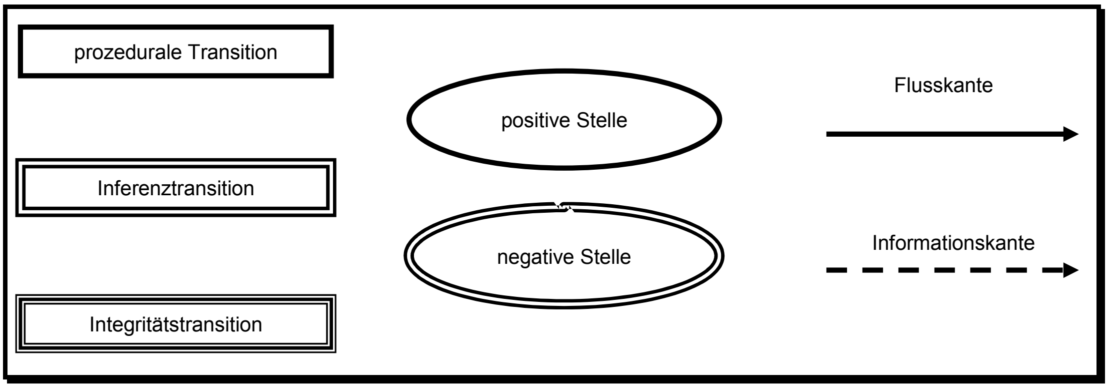

Abbildung 22: Symbole zur graphischen Visualisierung von Ontologie-Netzen

Prozedurale Transitionen werden durch einfache Rechtecke repräsentiert. Deklarative Transitionen werden hingegen durch ihre Linienart kenntlich gemacht. Während für Inferenztransitionen doppelt umrandete Rechtecke verwendet werden, werden für Integritätstransitionen dreifach umrandete Rechtecke benutzt. Darüber hinaus werden bei der graphischen Repräsentation von Stellen einfache runde Symbole für positive Stellen und zweifach umrandete runde Symbole für negative Stellen verwendet. Zudem wird eine Flusskante $(kn_x,kn_y)\in FR$ als Pfeil mit durchgezogener Linie dargestellt, der von dem Symbol, das den Ursprungsknoten kn_x repräsentiert, zu dem Symbol, das den Zielknoten kn_y repräsentiert, führt. Elemente der Informationsrelation IR werden analog zwar auch als Pfeile – allerdings mit gestrichelter Linie – dargestellt.

4.2.2 Struktur von Ontologie-Netzen

4.2.2.1 Statische Struktur von Ontologie-Netzen

4.2.2.1.1 Einfache statische Struktur von Ontologie-Netzen

Während in S/T-Netzen grundsätzlich lediglich anonyme Marken – streng genommen Kopien der einen unstrukturierten Basismarke – zugelassen sind, werden in Ontologie-

Netzen auch solche Marken zugelassen, die eine innere Struktur aufweisen. Somit sind die Marken in Ontologie-Netzen – im Gegensatz zu den Marken in S/T-Netzen – voneinander unterscheidbar. Dabei ist die Individualität von Marken in Ontologie-Netzen auf zwei unterschiedlichen Ebenen angesiedelt.

Zum einen handelt es sich bei der Individualität von Marken in Ontologie-Netzen um die *Unterscheidbarkeit* von Markenkopien[1)] bezüglich ihrer *Art*. Die Art einer Markenkopie entspricht der Konzeptfolge $w \in K^+$, der sie zugeordnet ist. Als Anfangsmarkierung einer Stelle st_m ist in einem Ontologie-Netz ON nur eine solche Multimenge $M_0(st_m) \in GTT_w$ von ontologischen Termtupeln zugelassen, deren Typ $w \in K^+$ mit dem Typ $typ_{RS_{OS}}(AN_{ST}(st_m))$ des Relationssymbols $AN_{ST}(st_m)=R_j$ übereinstimmt, mit dem die Stelle st_m annotiert ist. Insofern können unterschiedliche Stellen in Ontologie-Netzen auch unterschiedliche Arten von Marken aufweisen.[2)]

Zum anderen können in Ontologie-Netzen auch Marken gleicher Art voneinander unterschieden werden. Die Unterscheidbarkeit von Marken gleicher Art wird in Ontologie-Netzen darüber gewährleistet, dass Marken als Ausdrücke über einem ontologischen Alphabet $ALPH_{OS}$ konstruiert werden. Ununterscheidbar sind hingegen solche Markenkopien, mit denen eine Stelle st_m mehrfach markiert ist. Die Möglichkeit der mehrfachen Markierung von Stellen mit den gleichen Markenkopien wird durch die Eigenart jeder stellenbezogenen Markierung $M_r(st_m)$ eingeräumt, eine Multimenge zu sein.

Der Support eines Ontologie-Netzes ist die statische SIG_{ST}-Struktur $A_{SIG_{ST}}$, in der nur zeitlich invariante Extensionen zu Relationssymbolen aus SIG_{OS} berücksichtigt werden. Wie bereits zuvor aufgeführt, umfasst der Support $A_{SIG_{ST}}$ zu einem Ontologie-Netz ON eine Familie OBF_{OS} konzeptspezifischer Objektmengen, eine Familie OPF von Operationen und eine Familie RF_{ST} von Relationen. Mit den Mitgliedern $\{o_1,...,o_I\}$ der Operationenfamilie OPF werden *alle* Operationssymbole $\{O_1,...,O_I\}$ aus der ontologischen Signatur SIG_{OS} extensional interpretiert. Von den Relationssymbolen $\{R_1,...,R_J\} \in RS$ werden hingegen nur jene Relationssymbole aus SIG_{OS} durch Relationen r_j aus $A_{SIG_{ST}}$ extensional interpretiert, die zur Menge RS_{ST} aller statischen Relationssymbole gehören. Für die Relationssymbole aus der Menge RS_{DY} werden von $A_{SIG_{ST}}$ keine extensionalen Interpretationen angegeben.

---

1) Analog zu der Markierung von Stellen aus S/T-Netzen mit Kopien einer Basismarke werden die Stellen in Ontologie-Netzen mit Markenkopien markiert. Jede Marke entspricht in Ontologie-Netzen einem ontologischen Termtupel. Die Anzahl der Kopien einer Marke, mit denen eine Stelle st_m markiert ist, entspricht der Multiplizität des ontologischen Termtupels entsprechend der Multimenge, mit der st_m markiert ist. Die Unterscheidung zwischen Marken und ihren Kopien wird im Folgenden lediglich dann explizit angesprochen, wenn es für den Argumentationskontext notwendig oder hilfreich ist. Ansonsten wird – der üblichen Diktion folgend – von Marken gesprochen, wenn Markenkopien gemeint sind.

2) Allerdings reicht die potenzielle Unterschiedlichkeit der Markenarten nicht als hinreichendes Unterscheidungsmerkmal aus, da grundsätzlich auch solche Stellen in Ontologie-Netzen zugelassen sind, denen zwei voneinander unterschiedliche Relationssymbole zugewiesen sind, die allerdings die gleiche Typisierung aufweisen.

Die extensionale Interpretation von dynamischen Relationssymbolen geht mittelbar aus der Markierung eines Ontologie-Netzes hervor. Hierfür wird die Multimenge $M_z(st_m)$ ontologischer Grundtermtupel, mit der eine Stelle st_m in einem Zustand z markiert ist, mit Hilfe der entsprechenden konzeptfolgenspezifischen Termtupelauswertungsfunktion I_w zu einer Multimenge von Individuen aus OB ausgewertet. Die Multimenge von Individuen entspricht der Extension des dynamischen Relationssymbols R_j, das der Stelle st_m im Zustand z zugeordnet ist. Um die eineindeutige Beziehung zwischen zustandsbezogenen Markierungen und extensionalen Interpretationen von dynamischen Relationssymbolen zu gewährleisten, müssen den Stellen aus ON dynamische Relationssymbole aus SIG_{OS} und deren negative Pendants zugeordnet werden.

Die bijektive Zuordnung eines Relationssymbols $R_j \in (RS_{DY} \cup \underline{RS}_{DY})$ zu einer Stelle st_m erfolgt mit Hilfe der *Stellenannotationsfunktion*

$$AN_{ST}: ST \rightarrow RS_{DY} \cup \underline{RS}_{DY}.$$ [1)]

Die Stellenannotationsfunktion AN_{ST} weist jeder positiven Stelle $st_m \in ST_{POS}$ ein positives Relationssymbol $AN_{ST}(st_m)=R_j$ und jeder negativen Stelle $st_m \in ST_{NEG}$ ein negatives dynamisches Relationssymbol $AN_{ST}(st_m)=\underline{R}_j$ zu. Insofern gelten:

$$\forall st_m \in ST_{POS}: AN_{ST}(st_m) \in RS_{DY}$$
$$\text{und } \forall st_m \in ST_{NEG}: AN_{ST}(st_m) \in \underline{RS}_{DY}.$$

Bei AN_{ST} handelt es sich erstens um eine *linkstotale* Funktion. Demnach muss *jeder* Stelle $st_m \in ST$ über eine Annotation $AN_{ST}(st_m)$ ein Relationssymbol $AN_{ST}(st_m)=R_j$ oder $AN_{ST}(st_m)=\underline{R}_j$ zugeordnet werden. Zweitens ist die Stellenannotationsfunktion AN_{ST} injektiv.[2)] Es darf nämlich für jedes Relationssymbol $R_j \in (RS_{DY} \cup \underline{RS}_{DY})$ höchstens eine Stelle $st_m \in ST$ existieren, für die $AN_{ST}(st_m)=R_j$ gilt:

$$\forall st_1,st_2 \in ST, R_j \in (RS_{DY} \cup \underline{RS}_{DY}): (AN_{ST}(st_1)=R_j \wedge AN_{ST}(st_2)=R_j) \rightarrow st_1=st_2.$$

---

1) Um die korrekte Zuordnung von positiven und negativen Relationssymbolen zu positiven bzw. negativen Stellen zu gewährleisten, könnte eine „Splittung" der Stellenannotationsfunktion AN_{ST} zu den zwei Funktionen

$$AN_{ST_{POS}}: ST_{POS} \rightarrow RS_{DY} \text{ und}$$

$$AN_{ST_{NEG}}: ST_{NEG} \rightarrow \underline{RS}_{DY}$$

vorgenommen werden. Das hätte zwar den Vorteil der garantiert korrekten Zuordnung, allerdings müssten bei den folgenden Ausführungen stets die beiden Unterfälle positiver und negativer Stellen explizit unterschieden werden. Es wurde jedoch bereits früher darauf hingewiesen, dass die Differenzierung zwischen positiven und negativen Relationssymbolen einerseits und Stellen andererseits für Ontologie-Netze nicht notwendig ist. Die Differenzierung entfällt beispielsweise, wenn von einer „geschlossenen Weltmodellierung" und dem damit verbundenen Negation-by-failure-Prinzip ausgegangen wird. Die Korrektheit der Stellenannotation wird daher lediglich über zwei Integritätsbedingungen ausgedrückt, die weiter unten vorgestellt werden.

2) Vgl. JUOPPERI (1995), S. 28; KLEINJOHANN (1993), S. 60; TACKEN (2001), S. 42.

Dies ist notwendig, da die extensionale Interpretation von dynamischen Relationssymbolen mittelbar von der Markierung der jeweiligen Stelle angegeben wird.[1)] Würde ein dynamisches Relationssymbol $R_j \in (RS_{DY} \cup \underline{RS}_{DY})$ mehreren Stellen $st_1,...,st_n \in ST$ als Annotation zugewiesen werden, könnten die potenziell voneinander abweichenden Markierungen $M_0(st_1),...,M_0(st_n)$ der Stellen jeweils auch unterschiedlichen extensionalen Interpretationen der Relationssymbole entsprechen.[2)]

Darüber hinaus wird für konventionelle Pr/T-Netze in der Regel eingefordert, dass die Stellenannotationsfunktion AN_{ST} *surjektiv* und somit – in Kombination mit ihrer Injektivität – *bijektiv* sein müssen. Demnach würde in einem Ontologie-Netz ON für *jedes* positive oder negative dynamische Relationssymbol $R_j \in (RS_{DY} \cup \underline{RS}_{DY})$ genau eine Stelle $st_m \in ST$ existieren, für die $AN_{ST}(st_m)=R_j$ gilt. In Ontologie-Netzen braucht die Stellenannotationsfunktion AN_{ST} allerdings nicht surjektiv zu sein. Sie ist lediglich surjektiv bezüglich ihrer Teilmenge $ST \rightarrow RS_{DY}$. Für jedes positive dynamische Relationssymbol $R_j \in RS_{DY}$ wird nämlich im Ontologie-Netz genau eine positive Stelle vorgesehen, aus deren Markierung die Extension von R_j hervorgeht. Für negative dynamische Relationssymbole aus der Menge $\underline{RS}_{DY}$ muss jedoch nicht notwendig eine entsprechende negative Stelle im Ontologie-Netz existieren. Stellen zu negativen dynamischen Relationssymbolen werden nur dann in einem Ontologie-Netz ON benötigt, wenn explizit die Spezifikation *ungültiger* Formeln benötigt wird. Dies kann insbesondere im Vor- oder Nachbereich von deklarativen Transitionen der Fall sein.

---

1) Mit der Annotation $AN_{ST}(st_m)=R_j$ einer Stelle st_m wird angegeben, welche Markenarten auf der jeweiligen Stelle abgelegt werden dürfen. Dies ergibt sich im weiteren Verlauf aus dem Typ $typ_{RS_{DY}}(R_j)$, der für R_j definiert ist. Auf das Zusammenwirken von Stellenannotation einerseits und -markierung andererseits wird im weiteren Verlauf – in Form einer integritätsbewahrenden Formel – näher eingegangen.

2) Es kann jedoch durchaus sein, dass zwei unterschiedlichen Stellen st_1 und st_2 mit $st_1 \neq st_2$ die gleiche natürlichsprachliche *Bezeichnung*

$$bez_{lan_1}(AN_{ST}(st_1))=bez_{lan_2}(AN_{ST}(st_2))$$

für ihre Relationssymbole zugeordnet ist. In diesem Fall gilt aufgrund der Homonymie der metasprachlichen Relationssymbolbezeichnungen: $HOM(AN_{ST}(st_1))$ und $HOM(AN_{ST}(st_2))$. Handelt es sich bei den beiden Sprachen lan_1 um lan_2 um zwei unterschiedliche natürliche Sprachen, liegt eine Homonymie i.e.S. vor. Ansonsten liegt eine Homonymie i.w.S. vor. *Synonyme Stellenbezeichnungen i.e.S.* liegen hingegen genau dann vor, wenn einer Stelle st_m ein derartiges Relationssymbol $AN_{ST}(st_m)$ zugeordnet ist, dass

$$|bez_{lan}(AN_{ST}(st_m))| \geq 2$$

gilt. In diesem Fall wird einer Stelle st_m eine Menge $bez_{lan}(AN_{ST}(st_m))$ mit $SYN(bez_{lan}(AN_{ST}(st_m))$ von natürlichsprachlichen Bezeichnern zugeordnet, in der mindestens zwei metasprachliche Zeichenketten vorkommen. Die synonyme Stellenbezeichnung i.w.S. ist für eine Stelle st_m genau dann gegeben, wenn

$$bez_{lan_1}(AN_{ST}(st_m)) \cap bez_{lan_2}(AN_{ST}(st_m)) \neq \varnothing$$

mit $lan_1 \neq lan_2$ gilt.

Die Umkehrfunktion $AN_{RS}=AN_{ST}^{-1}$ ordnet jedem dynamischen Relationssymbol $R_j \in (RS_{DY} \cup \underline{RS}_{DY})$ diejenige Stelle $st_m \in ST$ zu, für die $AN_{ST}(st_m)=R_j$ gilt:

$$AN_{RS}\colon RS_{DY} \cup \underline{RS}_{DY} \rightarrow ST$$
$$\text{mit } \forall st_m \in ST,\ R_j \in (RS_{DY} \cup \underline{RS}_{DY})\colon AN_{ST}(st_m)=R_j \leftrightarrow AN_{RS}(R_j)=st_m.$$

Die *Anfangsmarkierungsfunktion*

$$M_0\colon ST \rightarrow MULT(GTT_{SIG_{OS}})$$

weist jeder Stelle $st_m \in ST$ eine Multimenge $mult \in MULT(GTT_{SIG_{OS}})$ von ontologischen Grundtermtupeln über der ontologischen Signatur SIG_{OS} als *stellenbezogene Anfangsmarkierung* $M_0(st_m)$ zu. Die Multimenge $M_0(st_m)$ gibt für jede Marke $<t_1,...,t_n>$ die für die Stelle st_m zulässig ist, die Anzahl der Kopien dieser Marke an, die sich unter der Anfangsmarkierungsfunktion M_0 auf der Stelle st_m befinden. Jede Marke entspricht wiederum einem ontologischen Grundtermtupel $<t_1...t_n>$. Entsprechend wird vorausgesetzt, dass es sich bei den Termen $t_1,...,t_n$, die in der stellenbezogenen Markierung vorkommen, um ontologische Grundterme handelt. Darüber hinaus muss das Grundtermtupel $<t_1,...,t_n>$, mit dem eine Stelle st_m markiert ist – entsprechend IB_{ON_3} – den gleichen Typ $w \in K^+$ haben, den auch das Relationssymbol $AN_{ST}(st_m)=R_j$ mit $typ_{RS_{DY}}(R_j)=w$ hat.

Aus dem Blickwinkel ihrer *Zusammengesetztheit* kann es sich bei den ontologischen Termen, aus denen eine stellenbezogene Markierung $M_0(st_m)$ konstruiert ist, um einfache oder zusammengesetzte Terme handeln. Beispielsweise ist eine stellenbezogene Markierung der Art

$$M_0(st_m)=<t_1,...,O_i(t_{x_1},...,t_{x_y}),...,t_n>$$

zulässig, wenn sie konform mit der Typisierung $typ_{RS_{OS}}(AN_{ST}(st_m))$ ist.[1)] Zudem sind aus dem Blickwinkel der *Wertigkeit* ontologischer Terme sowohl *einwertige* als auch *mengenwertige* Terme in den stellenbezogenen Markierungen zulässig.

Die Funktionsfamilie

$$MF_{ON}=(M_z)_{z \in ZR},\ ZR=\{0,...,Z\} \text{ mit } Z \in \mathbb{N}.$$

umfasst alle *möglichen* Markierungsfunktionen[2)]

$$M_z\colon ST \rightarrow MULT(GTT_{SIG_{OS}})$$

Der Zustandsraum ZR umfasst alle denkbaren Zustände z. Dabei ist nicht jede zustandsbezogene Markierung aus MF_{ON} auch zulässig. Für die Zulässigkeit einer zustandsbezogenen Markierung ist sind insbesondere die Integritätsbedingungen $\widehat{IB}_{ON_3}$ und $\widehat{IB}_{ON_4}$ von Bedeutung:

1) Auf die Konformität der stellenbezogenen Markierung $M_0(st_m)$ mit dem Typ $typ_{RS_{OS}}(AN_{ST}(st_m))$ wird weiter unten eingegangen.

2) Analog zu der Vorgehensweise bei S/T-Netzen wird auch bei Ontologie-Netzen die vereinfachte Formulierung *Markierung* zugelassen, wenn aus dem Kontext hervorgeht, dass es sich um eine Markierungsfunktion handelt.

$$\widehat{IB_{ON_3}} = \forall st_m \in ST, w \in K^+, M_z \in MF_{ON}:$$
$$typ_{RS_{ON}}(AN_{ST}(st_m)) = w \rightarrow M_z(st_m) \in MULT(GTT_w)$$

$$\widehat{IB_{ON_4}} = \forall st_m \in ST, M_z \in MF_{ON}: 0 \leq \#(M_z(st_m)) \leq KP(st_m).$$

Mit der Integritätsbedingung $\widehat{IB_{ON_3}}$ wird die Integritätsbedingung IB_{ON_3}, die nur auf die Anfangsmarkierung M_0 bezogen war, auf alle möglichen Markierungen erweitert. Eine mögliche Markierung $M_z \in MF_{ON}$ ist genau dann bezüglich $\widehat{IB_{ON_3}}$ zulässig, wenn sie nicht im Widerspruch zu $\widehat{IB_{ON_3}}$ steht. Dies ist genau dann der Fall, wenn unter M_z jede Stelle st_m mit einer Multimenge $M_z(st_m)$ markiert ist, die den gleichen Typ $w \in K^+$ hat wie das Relationssymbol $AN_{ST}(st_m)=R_j$. Die Integritätsbedingung $\widehat{IB_{ON_3}}$ ist notwendig, da die Extensionen von dynamischen Relationssymbolen aus den Markierungen der Stellen hervorgehen, denen sie zugewiesen sind. Entsprechend dürfen als Markierungen nur solche ontologischen Termtupel zugelassen werden, die konform mit dem Typen des Relationssymbols sind.

Analog stellt die Integritätsbedingung $\widehat{IB_{ON_4}}$ eine Erweiterung der Integritätsbedingung IB_{ON_4} dar. Eine mögliche Markierung $M_z \in MF_{ON}$ ist genau dann bezüglich $\widehat{IB_{ON_4}}$ zulässig, wenn die Kardinalität $\#(M_z(st_m))$ jeder zustands- und stellenbezogenen Markierung $M_z(st_m)$ weder negativ noch größer als die entsprechende Kapazität $KP(st_m)$ ist. Somit wird durch die stellenspezifische Kapazität $KP(st_m)$ eine *Obergrenze* für die Anzahl an Markenkopien festgelegt, mit denen st_m insgesamt markiert sein darf. Darüber hinausgehende Kapazitätsrestriktionen werden hingegen nicht berücksichtigt.[1)]

Die Funktionsfamilie

$$ZMF_{ON} \subset MF_{ON}$$

umfasst sämtliche zustandsbezogenen Markierungsfunktionen M_z aus MF_{ON}, die bezüglich $\widehat{IB_{ON_3}}$ und $\widehat{IB_{ON_4}}$ zulässig sind.

Mit Hilfe der Zuordnung von dynamischen Relationssymbolen zu Stellen und der stellenbezogenen Markierungen M_z durch Grundtermtupel können die zustandsbezogenen Extensionen von dynamischen Relationssymbolen angegeben werden. Aus diesem Blickwinkel entspricht jedes Grundtermtupel $<t_1...t_n>$, mit dem eine Stelle st_m mit $AN_{ST}(st_m)=R_j$ markiert ist, einem *Faktum* $R_j(t_1,...,t_n)$. Wir zuvor angesprochen, entspricht die Extension eines dynamischen Relationssymbols R_j in einem Zustand z derjenigen Multimenge von Individuen, die aus der Auswertung der stellenbezogenen Markierung $M_z(AN_{RS}(R_j))$ in dem Support $A_{SIG_{ST}}$ mit Hilfe einer konzeptfolgenspezifischen Termtupelauswertungsfunktion I_w hervorgeht. Dabei wird jene Termtupelauswertungs-

1) Als solche darüber hinausgehenden Kapazitätsrestriktionen kämen u.a. zwei Möglichkeiten in Frage. Bei der ersten Variante könnte eine Mindestanzahl an Marken festegelegt werden, mit denen eine Stelle markiert sein muss. Bei der zweiten Alternative könnte eine Maximalanzahl für Kopien derselben Marke auf einer Stelle angegeben werden; vgl. ZELEWSKI (1995), Bd. 5.1, S. 104. Für beide Erweiterungen der stellenspezifischen Markierungskardinalität gilt allerdings, dass sie für die Modellierung kooperativer Informationssysteme keine Rolle spielen. Sie werden daher im Folgenden nicht berücksichtigt.

funktion I_w verwendet, die der Konzeptfolge $typ_{RS_{OS}}(R_j) \in K^+$ zugeordnet ist. Somit gilt für die zustandsbezogene Extension $I_{RS_z}(R_j)$ eines dynamischen Relationssymbols R_j:

$$\forall R_j \in RS_{DY},\ st_m \in ST,\ M_z \in MF_{ON}:$$
$$AN_{ST}(st_m)=R_j \rightarrow I_{RS_z}(R_j) = I_w(M_z(st_m)).$$

Bei den stellenbezogenen Markierungen aus einem Ontologie-Netz ON handelt es sich stets um Multimengen ontologischer *Grund*termtupel. Entsprechend bedarf es bei der Auswertung von Markierungen keiner Variablenbelegung. Als Rekursionsbasis für die Auswertung von stellenbezogenen Markierungen kommt somit lediglich die Auswertung von Konstantensymbolen mit Hilfe einer extensionalen Interpretationsfunktion I_{OPS} in Frage. Die Auswertung von Konstantensymbolen kann wiederum in dem Support $A_{SIG_{ST}}$ erfolgen, da für alle Operationssymbole – und somit auch für Konstantensymbole – zustandsinvariante Extensionen vorausgesetzt wurden. Somit können die Terme in den stellenbezogenen Markierungen mit Hilfe der extensionalen Interpretationsfunktion I_{OPS} aus $A_{SIG_{ST}}$ zu Individuen aus dem Objektbereich OB von $A_{SIG_{ST}}$ ausgewertet werden. Beispielsweise kann die Markierung

$$M_z(st_m)=<t_{1_1},...,t_{1_Y}>+...+<t_{X_1},...,t_{X_Y}>$$

einer Stelle st_m wie folgt zu einer Multimenge von Objekttupeln ausgewertet werden, wenn sie nur aus Konstantensymbolen zusammengesetzt ist:

$$I_w(<t_{1_1},...,t_{1_Y}>+...+<t_{X_1},...,t_{X_Y}>) \in (OB_{k_1} \times ... \times OB_{k_n} \rightarrow \mathbb{N})$$
$$= I_w(<t_{1_1},...t_{1_Y}>) \oplus ... \oplus I_w(<t_{X_1},...t_{X_Y}>)$$
$$\text{mit } I_w(<t_{x_1},...,t_{x_Y}>)=(ob_{x_1},...,ob_{x_Y})$$
$$\text{und } ob_{x_y}=IT_{k_x}(t_{x_y})=I_{OPS}(t_{x_y})=o_{ix_y}() \text{ für alle } x=1,...,X \text{ und } y=1,...,Y.$$

Wenn in der Markierung $<t_{1_1},...,t_{1_Y}>+...+<t_{X_1},...,t_{X_Y}>$ auch zusammengesetzte ontologische Terme vorkommen, erfolgt ihre Auswertung entsprechend dem rekursiven Schema der konzeptspezifischen Termauswertung. Auch hierbei reicht die extensionale Interpretationsfunktion I_{OPS} aus, da variable Terme nicht vorkommen und somit nicht auf eine Variablenbelegung zurückgegriffen werden muss. Im Fall eines zusammengesetzten Terms $O_i(t_{x_1},...,t_{x_n})$ wird die Operation $o_i=I_{OPS}(O_i)$, die aus der Auswertung des Operationssymbols O_i hervorgeht, auf die Auswertungen der Terme $t_{x_1},...,t_{x_n}$ im Argument des zusammengesetzten Terms angewandt. Auch hierbei kann es sich wieder um zusammengesetzte Terme handeln, die jedoch in ihren Basen stets als atomare Terme nur Konstantensymbole aufweisen dürfen. Letztgenannte werden wieder durch die Interpretationsfunktion I_{OPS} zu Konstanten ausgewertet.

Ein bemerkenswerter Fall – durch dessen Zulässigkeit Ontologie-Netze von vielen anderen höheren Petri-Netz-Klassen abgegrenzt werden können – tritt dann ein, wenn mindestens einer der Terme aus der stellenbezogenen Markierung $M_0(st_m)$ mengenwertig ist. In diesem Fall muss der entsprechende Term zu einem mengenwertigen Individuum aus dem Support $A_{SIG_{ST}}$ ausgewertet werden. Wenn beispielsweise eine stellenbezogene Markierung der Art

$$M_0(st_m)=<t_1,...,t_n>$$

vorliegt, bei der nur[1)] der ontologische Grundterm $t_n \in MT_{k_n}$ mengenwertig ist, muss die konzeptfolgenspezifische Termtupelauswertung lauten:

$I_w(<t_1,...,t_n>)$ mit $t_x \in TERM_{k_x}$, x=1,...,n und $k_1,...,k_{n-1} \in K_{EW}$ und $k_n \in K_{MW}$
$\in (OB_{k_1} \times ... \times OB_{k_n} \rightarrow \mathbb{N})$,
mit $I_w(<t_1,...,t_{x_Y}>) = (ob_1,...,ob_n) = (ob_1,...,ob_{n-1},\{ob_{n_1},...,ob_{n_m}\})$
und $ob_x = IT_{k_x}(t_x) = ob_x$ mit $ob_n = \{ob_{n_1},...,ob_{n_m}\}$ für x=1,...,n.

In den betrachteten Beispielen werden aus Gründen der Einfachheit lediglich stellenbezogene Markierungen ausgewertet, in denen eine Höchstmultiplizität von 1 für jedes Termtupel bestimmt ist. Genauso lassen sich allerdings auch Markierungen auswerten, in denen Termtupel mit Multiplizitäten größer als 1 vorkommen.

Die *Transitionsannotationsfunktion*

$$AN_{TR}: TR \rightarrow FORM_{SIG_{ST}}$$

weist jeder Transition $tr_n \in TR$ eine Formel $F \in FORM_{SIG_{ST}}$ über der statischen ontologischen Signatur SIG_{ST} als Annotation $AN_{TR}(tr_n)=F$ zu. Mit der Annotation $AN_{TR}(tr_n)$ einer Transition $tr_n \in TR$ durch die Formel $F \in FORM_{SIG_{ST}}$ wird eine Bedingung formuliert, die bei der Überprüfung von Schaltvoraussetzungen für tr_n eine Rolle spielt.

Bei der Transitionsannotationsfunktion AN_{TR} handelt es sich um eine totale Funktion. Das bedeutet, dass zu jeder Transition $tr_n \in TR$ aus einem Ontologie-Netz ON genau eine statische ontologische Formel $F \in FORM_{SIG_{ST}}$ existieren muss, die die Annotation $AN_{TR}(tr_n)=F$ von tr_n ist. Denn die Schaltvoraussetzungen für Transitionen aus Ontologie-Netzen hängen zum einen von den Markierungen der Stellen in der Nachbarschaft der Transitionen und zum anderen von den Transitionsannotationen ab. Die Markierungen der Nachbarschaftsstellen $st_m \in NB_{TR}(tr_n)$ einer Transition tr_n sind die *dynamischen* Schaltvoraussetzungen für tr_n. Die Transitionsannotation $AN_{TR}(tr_n)$ stellen hingegen die *statischen* Schaltvoraussetzungen dar. In Fällen, in denen das Schalten der Transition tr_n lediglich von den Markierungen der adjazenten Stellen abhängen soll, wird tr_n mit der tautologischen Formel $w \in FORM_{SIG_{ST}}$ annotiert. Dadurch, dass die statische Schaltvoraussetzung hier immer erfüllt ist, werden in diesem Fall für tr_n nur dynamische Schaltvoraussetzungen betrachtet. Transitionen, die hingegen mit der kontradiktorischen

1) Aus Gründen der Einfachheit wird im Beispiel eine stellenbezogenen Markierung aufgezeigt, in der

(1.) nur ein Termtupel $<t_1,...,t_n>$ vorkommt und

(2.) nur der Term $t_n \in TERM_{k_n}$ mit $k_n \in K_{MW}$ mengenwertig ist.

Formel $f \in FORM_{SIG_{OS}}$ annotiert werden, können grundsätzlich nicht schalten, da bereits die statische Schaltvoraussetzung nicht erfüllt ist.[1)]

Als Annotation $AN_{TR}(tr_n)$ für eine Transition $tr_n \in TR$ werden nur statische ontologische Formeln aus der Menge $FORM_{SIG_{ST}}$ zugelassen. Es wurde bereits darauf hingewiesen, dass für die Konstruktion von statischen ontologischen Formeln nur statische Relationssymbole aus der Menge RS_{ST} herangezogen werden dürfen. Entsprechend können die Annotationen $AN_{TR}(tr_1),...,AN_{TR}(tr_N)$ im Ontologie-Netz ON in dem Support $A_{SIG_{ST}}$ ausgewertet werden. Dabei wird zunächst offen gelassen, ob die Annotation $AN_{TR}(tr_n)$ zu einer Transition $tr_n \in TR$ freie Variablen enthält.

Mit der *allgemeinen Flusskantenannotation*

$$AN_{FR}: ((ST \times TR) \cup (TR \times ST)) \rightarrow MULT(TT_{SIG_{OS}})$$

wird jedes Element $(kn_x,kn_y) \in ((ST \times TR) \cup (TR \times ST))$ auf eine Multimenge $AN_{FR}(x,y) \in MULT(TT_{SIG_{OS}})$ von Termtupeln über der ontologischen Signatur SIG_{OS} abgebildet. Zur Bestimmung der Bilder der allgemeinen Flusskantenannotation AN_{FR} wird die *spezielle Flusskantenannotation*

$$AN^*_{FR}: FR \rightarrow MULT(TT_{SIG_{OS}})$$

verwendet. Im Gegensatz zu der allgemeinen Flusskantenannotation AN_{FR} nimmt die spezielle Flusskantenannotation AN^*_{FR} in ihrem Argument nur solche aus zwei artverschiedenen Knoten kn_x und kn_y bestehenden Tupel $(kn_x,kn_y) \in ((ST \times TR) \cup (TR \times ST))$ auf, die auch in der Flussrelation FR enthalten sind. Die Bilder der allgemeinen Flusskantenannotation AN_{FR} können mit Hilfe der speziellen Flusskantenannotation AN^*_{FR} definiert werden als:[2)]

---

1) Darüber hinaus muss die Transitionsannotationsfunktion AN_{TR} nicht *injektiv* sein. Zwei unterschiedlichen Transitionen $tr_1,tr_2 \in TR$ kann die gleiche statische ontologische Formel $F \in FORM_{SIG_{OS}}$ als Annotation $AN_{TR}(tr_1)=AN_{TR}(tr_2)=F$ zugewiesen sein. Dies ist beispielsweise dann der Fall, wenn als Schaltbedingung mehrerer Transitionen in einem Ontologie-Netz nur die Markierungen adjazenter Stellen berücksichtigt werden sollen und sie daher mit $w \in FORM_{SIG_{ST}}$ annotiert werden.
Auf jeden Fall ist die Transitionsannotationsfunktion AN_{TR} nicht surjektiv. Im Fall der Surjektivität würde nämlich für jede statische ontologische Formel $F \in FORM_{SIG_{ST}}$ eine Transition $tr \in TR$ mit $AN_{TR}(tr)=F$ existieren. Da in der Menge $FORM_{SIG_{ST}}$ auch widersprüchliche ontologische Formeln enthalten sind, ist AN_{TR} nicht surjektiv. Die Annotation einer Transition tr mit einer widersprüchlichen ontologischen Formel $F \in FORM_{SIG_{ST}}$ würde nur dann Sinn machen, wenn die Transition niemals schalten können soll. Für solche Fälle wird allerdings die kontradiktorische Formel $f \in FORM_{SIG_{ST}}$ verwendet, deren formale Semantik mit der formalen Semantik von widersprüchlichen ontologischen Formeln übereinstimmt.

2) Vgl. RÖBBECKE (1995), S. 64.

$$AN_{FR}(kn_x,kn_y)=\begin{cases} AN^*_{FR}(kn_x,kn_y); & \text{falls } (kn_x,kn_y)\in FR \\ \varnothing_M; & \text{sonst.}\end{cases}$$

Analog zu der allgemeinen und speziellen Gewichtungsfunktion G bzw. G* aus S/T-Netzen vereinfacht die Unterscheidung zwischen allgemeiner und spezieller Kantenannotation AN_{FR} bzw. AN^*_{FR} die Formulierung von Schaltvoraussetzungen für Transitionen aus Ontologie-Netzen.

Zum gleichen Zweck wird die allgemeine Informationskantenannotation AN_{IR} auf die spezielle Informationskantenannotation

$$AN^*_{IR}: IR \rightarrow MULT(TT_{SIG_{OS}})$$

zurückgeführt:

$$AN_{IR}(st_m,tr_n)=\begin{cases} AN^*_{IR}(st_m,tr_n); & \text{falls } (st_m,tr_n)\in IR \\ \varnothing_M; & \text{sonst.}\end{cases}$$

Für die Kantenannotationen AN_{FR} und AN_{IR} sind die Integritätsbedingungen $IB_{ON_{1.1}}$ und $IB_{ON_{1.2}}$ bzw. IB_{ON_2} von Relevanz. Den Integritätsbedingungen $IB_{ON_{1.1}}$ und $IB_{ON_{1.2}}$ zufolge muss die Konzeptfolge $w\in K^+$, der die Flusskantenannotation $AN_{FR}(st_m,tr_n)$ oder $AN_{FR}(tr_n,st_m)$ einer Flusskante $(st_m,tr_n)\in FR$ bzw. $(tr_n,st_m)\in FR$ zugeordnet ist, mit dem Typen $typ_{RS_{DY}}(R_j)$ desjenigen Relationssymbols $R_j=AN_{ST}(st_m)$ übereinstimmen, mit dem die Stelle st_m annotiert ist. Das gleiche gilt – entsprechend IB_{ON_2} – für Kanten aus der Informationsrelation IR. Für die Kantenannotationen gilt, dass auch die leere Multimenge $\varnothing_M$ als Kantenannotation zugewiesen werden kann. Da die leere Multimenge $\varnothing_M$ ein Element jeder Menge $MULT(TT_w)$ ist, ist $\varnothing_M$ als Fluss- bzw. Informationskantenannotation entsprechend $IB_{ON_{1.1}}$ und $IB_{ON_{1.2}}$ bzw. IB_{ON_2} für jede Kante zulässig.

Mit den Integritätsbedingungen $IB_{ON_{1.1}}$ und $IB_{ON_{1.2}}$ wird gewährleistet, dass nur solche Flusskantenannotationen spezifiziert werden dürfen, die von der adjazenten Stelle st_m Marken ab- bzw. zufließen lassen können, weil sich die Flusskantenannotation mit der Typisierung von st_m deckt. Dabei gilt für Ontologie-Netze – im Vergleich zu konventionellen Pr/T-Netzen – eine Besonderheit. Sie liegt darin, dass auch Flusskanten spezifiziert werden können, die mit solchen Multimengen ontologischer Termtupel annotiert sein können, aufgrund derer *nicht jede* Marke von der adjazenten Stelle abgezogen bzw. auf dieser abgelegt werden kann. Die Marken können in Ontologie-Netzen „gefiltert" abgezogen werden.[1)] Eine Multimenge $MULT(TT_{w_1})$ über einer Menge TT_{w_1} w_1-spezifischer Termtupel umfasst nämlich auch Multimengen solcher ontologischen

1) Es wird deutlich darauf hingewiesen, dass der „Filtermechanismus" aus Ontologie-Netzen keine Gemeinsamkeiten zu den *Filterkanten* aus NR/T-Netzen aufweist. Filterkanten werden in NR/T-Netzen dazu verwendet, zustandsändernde Grundoperationen auf „Subterme" von stellenbezogenen Markierungen auszuüben; vgl. OBERWEIS (1996), S. 129 ff.; OBERWEIS/SANDER (1996), S. 305 ff.

Termtupel, die zwar originär einer Konzeptfolge w_2 zugeordnet sind, allerdings derivativ auch der Konzeptfolge w_1 zugeordnet werden können, weil

$$w_2 \preceq w_1$$

gilt. Somit wird in Ontologie-Netzen grundsätzlich die Möglichkeit eingeräumt, solche Kantenannotationen zu spezifizieren, deren Typ nicht originär, aber derivativ mit dem Typ der adjazenten Stelle übereinstimmt.[1)] Dies wird im Folgenden anhand eines Beispiels verdeutlicht.

Die Teilspezifikationen der zugrunde gelegten Ontologie $SPEZ_{OS}$ lauten:[2)]

---

1) Üblicherweise werden für Pr/T-Netze die Integritätsregeln

$$\forall(st_m,tr_n)\in(ST\times TR)\text{: } typ_{RS_{OS}}(AN_{ST}(st_m)) = w_1 \wedge AN_{FR}(st_m,tr_n)\in MULT(TT_{w_2}) \rightarrow w_2 = w_1$$

und

$$\forall(tr_n,st_m)\in(TR\times ST)\text{: } typ_{RS_{OS}}(AN_{ST}(st_m)) = w_1 \wedge AN_{FR}(st,tr_n)\in MULT(TT_{w_2}) \rightarrow w_2 = w_1.$$

aufgestellt; vgl. JUOPPERI (1995), S. 28. Demnach müssen in Pr/T-Netzen die Typen der Kantenannotationen mit den Typen der jeweiligen adjazenten Stellenannotation *übereinstimmen*.

2) Es werden hier lediglich jene Komponenten aufgeführt, die für die Annotation von Kanten relevant sind. Darüber hinausgehende Aspekte werden im Rahmen der späteren Fallstudie ausführlicher behandelt.

K

{Mitarbeiter, Wissenschaftlicher_Mitarbeiter, Administrativer_Mitarbieter, Real};

$\sqsubseteq$

Wissenschaftlicher_Mitarbeiter $\sqsubseteq$ Mitarbeiter,

Administrativer_Mitarbieter $\sqsubseteq$ Mitarbeiter;

OPS

MULTIP Real Real $\rightarrow$ Real;

RS

RS_{ST}

Equal <⊤ ⊤>;

RS_{DY}

Mitarbeitergehalt <Mitarbeiter Real>;

$VARF_{SIG_{OS}}$

$VAR_{Mitarbeiter}$ $=\{MA\}$,

$VAR_{Wissenschaftlicher\_Mitarbeiter}$ $=\{WMA\}$,

$VAR_{Administrativer\_Mitarbeiter}$ $=\{AMA\}$,

VAR_{Real} $=\{G_{Alt}, G_{Neu}\}$;

$AN_{ST}(st_m)$=Mitarbeitergehalt.

Mit den Ausdrucksmitteln der Ontologie $SPEZ_{OS}$ wird ein Ontologie-Netz ON entsprechend Abbildung 23 konstruiert.

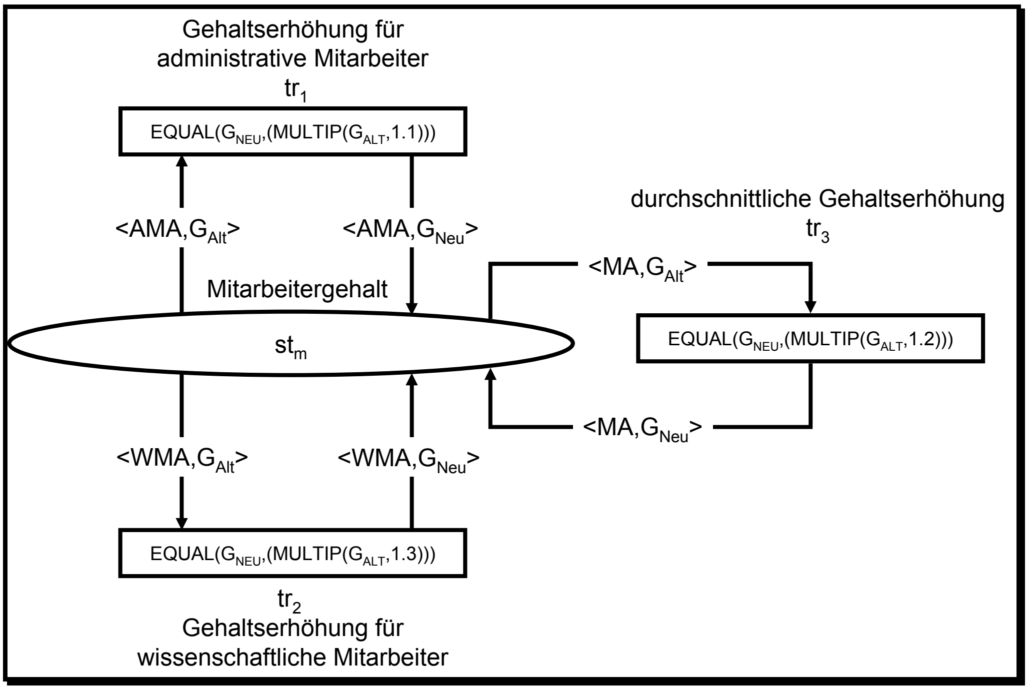

Abbildung 23: Gefilterte Kantenannotation in Ontologie-Netzen

Auf der Stelle st_m sind Markenkopien enthalten, mit denen die Gehälter von Mitarbeitern repräsentiert werden. Darunter befinden sich sowohl administrative als auch wissenschaftliche Mitarbeiter. Die beiden Transitionen tr_1 und tr_2 werden verwendet, um Gehaltserhöhungen für administrative bzw. wissenschaftliche durchzuführen. Mit der Transition tr_3 wird hingegen das Gehalt unabhängig von der Anstellung erhöht.

Sämtliche Kantenannotationen im betrachteten Ontologie-Netz sind bezüglich der Integritätsbedingungen $IB_{ON_{1.1}}$ und $IB_{ON_{1.2}}$ zulässig. Durch die unterschiedlichen Kantenannotationen wird gewährleistet, dass die Transitionen von st_m nur solche Marken abziehen bzw. auf st_m ablegen können, die für die jeweilige Transition erlaubt sind. Beispielsweise kann[1)] die Transition tr_1 nur solche Marken von der Stelle st_m abziehen, deren erster Term einem administrativen Mitarbeiter entspricht, da die Variable *AMA* in der Kantenannotation $AN_{FR}(st_m,tr_1)$ dem Konzept *Administrativer_Mitarbeiter* zugeordnet ist. Entsprechend kann die Transition tr_2 nur solche Marken von der Stelle st_m abziehen, deren erster Term einem wissenschaftlichen Mitarbeiter entspricht, da die Variable *WMA* in der Kantenannotation $AN_{FR}(st_m,tr_2)$ dem Konzept *Wissenschaftlicher_Mitarbeiter* zugeordnet ist. Die Kantenannotation $AN_{FR}(st_m,tr_3)$ weist in ihrer ersten Stelle die Variable MA auf, welche dem Konzept *Mitarbeiter* zugeordnet ist. Dadurch kann tr_3 *jede* beliebige Marke von der Stelle st_m abziehen. Denn sowohl das Konzept *Adminisrativer_Mitarbeiter* als auch das Konzept *Wissenschaftlicher_Mitarbeiter* sind dem Kon-

1) Der Markenabzug ist zunächst nur potenziell bestimmt, da für den faktischen Markenabzug auch die Substitution berücksichtigt werden muss, die beim Schaltakt unterstellt wird. Auf diesen Aspekt wird später näher eingegangen.

zept *Mitarbeiter* durch die Strukturierungsrelation $\sqsubseteq$ taxonomisch untergeordnet. Insofern gelten:

$TERM_{Administrativer\_Mitarbeiter} \subseteq TERM_{Mitarbeiter} \supseteq TERM_{Wissenschaftlicher\_Mitarbeiter}$ und
$GTT_{Administrativer\_MitarbeiterReal} \subseteq GTT_{MitarbeiterReal} \supseteq TERM_{Wissenschaftlicher\_MitarbeiterReal}$.

Mit der Möglichkeit, auch solche Kanten spezifizieren zu können, deren originärer Typ dem Typ der adjazenten Stelle untergeordnet ist, werden vielfältige Modellierungsmöglichkeiten für Ontologie-Netze erschlossen. Der Typ $w \in K^+$ einer Kante $(st_m,tr_n) \in (ST \times TR)$ mit $AN_{FR}(st_m,tr_n) \in MULT(TT_w)$ bestimmt nämlich, welche Art von Marken beim Schalten der adjazenten Transition tr_n von der adjazenten Stelle st_m abgezogen werden kann. Alternativ gibt der Typ $w \in K^+$ einer Kante $(tr_n,st_m) \in (TR \times ST)$ mit $AN_{FR}(tr_n,st_m) \in MULT(TT_{SIG_w})$ vor, welche Art von Marken beim Schalten der adjazenten Transition tr_n auf der adjazenten Stelle st_m abgelegt werden kann. Beide Aspekte werden deutlicher, wenn in den folgenden Abschnitten die dynamische Struktur von Ontologie-Netzen vorgestellt wird.

Damit die Kantenannotation AN_{FR} in einem Ontologie-Netz ON korrekt ist, darf keine Ausgangskante $(tr_n,st_1) \in (TR \times ST)$ zu einer Transition $tr_n \in TR$ mit einer Annotation $AN_{FR}(tr_n,st_1)$ existieren, in der auch Variablen vorkommen, die weder in mindestens einer Annotation $AN_{FR}(tr_n,st_2)$ einer Eingangskante (st_2,tr_n) zu tr_n noch in der Annotation $AN_{TR}(tr_n)$ vorkommen. Um die Korrektheit der Kantenannotation überprüfen zu können, werden die *Eingangsvariablenfunktion* var_{IN} und die *Ausgangsvariablenfunktion* var_{OUT} eingeführt.

Mit der Eingangsvariablenfunktion

$$var_{IN}: TR \rightarrow pot(VAR)$$
$$\text{mit } var_{IN}(tr_n) = \cup_{st_m \in VB_{TR}(tr_n)} var_{TT}(AN_{FR}(st_m,tr_n))$$

wird die Menge $var_{IN}(tr_n)$ der Variablen bestimmt, die in mindestens einer der Annotationen $AN_{FR}(st_1,tr_n),...,AN_{FR}(st_m,tr_n)$ der in die Transition tr_n eingehenden Kanten $(st_1,tr_n),...,(st_m,tr_n) \in (FR \cap (ST \times TR))$ vorkommen.[1)]

Mit der *Ausgangsvariablenfunktion*

$$var_{OUT}: TR \rightarrow pot(VAR)$$
$$\text{mit } var_{OUT}(tr_n) = \cup_{st_m \in NB_{TR}(tr_n)} var_{TT}(AN_{FR}(tr_n,st_m))$$

wird die Menge $var_{OUT}(tr_n)$ der Variablen erfasst, die in mindestens einer Annotationen $AN_{FR}(tr_n,st_1),...,AN_{FR}(tr_n,st_n)$ der von der Transition tr_n ausgehenden Kanten $(tr_n,st_1),...,(tr_n,st_n) \in (FR \cap (TR \times ST))$ vorkommen.[2)]

Die Annotation $AN_{FR}(tr_n,st_1)$ einer Ausgangskante $(tr_n,st_1) \in (TR \times ST)$ zu einer Transition $tr_n \in TR$ darf nur solche Variablen umfassen, die in mindestens einer Annotation

1) Vgl. TACKEN (2001), S. 44.

2) Vgl. TACKEN (2001), S. 44.

($AN_{FR}(st_m,tr_n)$ einer Eingangskante $(st_m,tr_n)\in(ST\times TR)$ zu tr_n oder der Transitionsannotation $AN_{TR}(tr_n)$ vorkommt:[1)]

$$\forall x\in VAR,\ tr_n\in TR:\ x\in var_{OUT}(tr_n) \rightarrow (x\in var_{IN}(tr_n) \vee x\in var_{F_{OS}}(AN_{TR}(tr_n))).$$

4.2.2.1.2 Erweiterte statische Struktur von Ontologie-Netzen

4.2.2.1.2.1 Inferenztransitionen

In Abschnitt 3.1.4.2 wurde vorgestellt, wie auf der Basis einer Ontologie $SPEZ_{OS}$ eine ontologiegestützte Wissensbasis WBS konstruiert werden kann. Eine wesentliche Komponente war hierbei, dass einerseits mit Hilfe der Inferenzregeln aus der expliziten Faktenbasis FB_{EXPL} inferenziell alle Fakten erschlossen werden können, die darin implizit enthalten sind. Andererseits wurde aufgezeigt, wie die Zulässigkeit von wissensrepräsentierenden Fakten mit Hilfe von Integritätsregeln überprüft werden kann. Beide Verfahren basierten auf den Prinzipien der modelltheoretischen Semantik, anhand derer Ontologien extensional interpretiert werden können.

In Ontologie-Netzen kann allerdings nicht immer über diese modelltheoretische Semantik verfügt werden. Denn in einem Ontologie-Netz ON ist lediglich die extensionale Semantik für statische Relationssymbole unmittelbar enthalten. Sie wird in Form der statischen SIG_{ST}-Struktur $A_{SIG_{ST}}$ angegeben, die Bestandteil jedes Ontologie-Netzes ON ist. An die SIG_{ST}-Struktur $A_{SIG_{ST}}$ wurde die Anforderung gestellt, dass alle Inferenz- und Integritätsregeln, die nur mit statischen Relationssymbolen konstruiert sind, in $A_{SIG_{ST}}$ gültig sind. Die Extensionen von dynamischen Relationssymbolen gehen lediglich mittelbar aus den stellenspezifischen Markierungen hervor. Entsprechend können Inferenz- und Integritätsregeln, in denen dynamische Relationssymbole vorkommen, in $A_{SIG_{ST}}$ nicht ausgewertet werden.

Je nachdem, welche zustandsspezifische Markierung vorliegt, kann es sein, dass die Bedingungen für eine Inferenzregel erfüllt sind. Solange keine Verfahren zur Berücksichtigung von Inferenzregeln in Ontologie-Netzen formuliert sind, können keine Fakten abgeleitet werden, die aus der Anwendung von Inferenzregeln auf eine explizite Faktenbasis hervorgehen. Entsprechend können keine Inferenzen auf der Basis der modelltheoretischen Semantik durchgeführt werden, wenn die entsprechenden Inferenzregeln u.a. mit Hilfe von dynamischen Relationssymbolen konstruiert sind.

Zudem bleiben in der bisherigen Definition von Ontologie-Netzen Integritätsregeln unberücksichtigt, die u.a. mit Hilfe von dynamischen Relationssymbolen konstruiert werden. Um die Zulässigkeit von Markierungen bezüglich Integritätsregeln überprüfen zu

1) Vgl. TACKEN (2001), S. 44, mit der Erweiterung um „Anweisungsketten“, mittels derer die Variablensubstitution bereits in der Transitionsannotation erfolgt.

können, bedarf es eines Verfahrens zur Berücksichtigung von Integritätsregeln in Ontologie-Netzen.

Um Inferenz- und Integritätsregeln in Ontologie-Netzen berücksichtigen zu können, werden im Folgenden zwei Verfahren vorgestellt, mit denen die deklarativen Regeln in operationale Ontologie-Netz-Komponenten transformiert werden können.[1] Das erste Verfahren dient dazu, auf Basis der in $SPEZ_{OS}$ enthaltenen Inferenzregeln ein solches Ontologie-Teilnetz TN zu erzeugen, mit dessen Hilfe „Inferenzen“ durchgeführt werden können. Im Gegensatz zu den Inferenzen, die im Rahmen der modelltheoretischen Semantik durchgeführt werden, handelt es sich hierbei allerdings um solche Inferenzen, die auf der operationalen Semantik von Petri-Netzen basieren. Die Inferenzen werden nämlich durch das Schalten von Inferenztransitionen durchgeführt. So erfolgt eine explizite Unterscheidung von Zuständen, zwischen denen der Schaltakt einer Inferenztransition liegt, die aus der Transformation einer Inferenzregel hervorgegangen ist. Das zweite Verfahren dient dazu, ein solches Ontologie-Teilnetz TN zu erzeugen, mit dessen Hilfe die Zulässigkeit einer zustandspezifischen Markierung bezüglich einer Integritätsregel überprüft werden kann. Beide Verfahren werden in Form von Algorithmen operationalisiert, die als Eingangswert eine Inferenz- bzw. Integritätsregel aufnehmen und als Ausgangswert ein Ontologie-Teilnetz TN umfassen. Somit ist die Transformation von Inferenz- und Integritätsregeln in Ontologie-Teilnetze durchgehend formal bestimmt. Dabei wird vorausgesetzt, dass alle zu transformierenden Inferenz- und Integritätsregeln den syntaktischen Anforderungen an Inferenz- und Integritätsregeln genügen.[2]

Eine Voraussetzung für die algorithmische Transformierbarkeit einer Inferenzregel

$$F = \forall x_1,...,x_k\!: (lit_1 \wedge .... \wedge lit_n) \rightarrow lit_{n+1}$$

in ein Ontologie-Teilnetz TN ist die Existenz einer regelspezifischen *Inferenztransition* $trans_{INF}(F)=tr$ mit $tr \in TR_{INF}$. Vor der Transformation wird die Annotation $AN_{TR}(tr_n)$ der Inferenztransition tr_n mit der tautologischen Formel $w \in FORM_{SIG_{ST}}$ initialisiert. Bei der Transformation werden zunächst alle Stellen, die den dynamischen Relationssymbolen aus den Antezedenz-Literalen $lit_1,...,lit_n$ von F zugeordnet sind, über Informationskanten mit tr verbunden. Handelt es sich hierbei um ein positives Literal der Form $R_j(t_{j1},...,t_{jl(j)})$, so wird diejenige Stelle $st_m \in ST_{POS}$ verwendet, die dem positiven Relationssymbol R_j durch $AN_{RS}(R_j)=st_m$ zugeordnet ist. Handelt es sich hingegen um ein negatives Literal der Form $\neg R_j(t_{j1},...,t_{jl(j)})$, so wird diejenige negative Stelle $st_m \in ST_{NEG}$ verwendet, die dem negativen Relationssymbol $\underline{R}_j$ durch $AN_{RS}(\underline{R}_j)=st_m$ zugeordnet ist. Die Annotationen der Informationskanten, die von der Stelle st_m ausgehen und in die Inferenztransition tr_n eingehen, werden jeweils mit dem Termtupel $<t_{j1},...,t_{jl(j)}>$ erweitert, das mit dem

1) Zur Transformation deklarativer Regeln in Transitionen in Petri-Netzen vgl. DAHR (1994), S. 19 ff.; DE/MAINDREVILLE/SIMON (1987), S. 11 ff.; LAUTENBACH (2003), S. 276 ff.; LIN/CHANSON (1998), S. 825 ff.; ZELEWSKI (1995), Bd. 5.1, S. 194 ff.

2) Vgl. Abschnitt 3.1.4.1.

Argument $(t_{j_1},...,t_{j_{l(j)}})$ des Literals j mit j=1,...,n übereinstimmt. Wenn ein Antezedenz-Literal hingegen mit Hilfe von statischen Relationssymbolen konstruiert ist, dann wird die Annotation $AN_{TR}(tr_n)$ der Inferenztransition tr_n entweder mit diesem Literal identifiziert oder um dieses Literal konjunktiv erweitert.[1)]

Bezüglich des Konklusions-Literals der Inferenzregel wird vorausgesetzt, dass das Literal mit Hilfe eines dynamischen Relationssymbols konstruiert ist. Denn ansonsten könnte die Extension des Relationssymbols nicht netzendogen variiert werden.[2)] Auch hierbei wird zwischen dem positiven und dem negativen Fall unterschieden. Handelt es sich bei dem Konklusions-Literal um ein positives Literal der Form $R_j(t_{j_1},...,t_{j_{l(j)}})$, so wird die Flussrelation FR um die Kante (tr_n,st_m) erweitert, wobei $st_m \in ST_{POS}$ mit $st_m=AN_{RS}(R_j)$ der positiven Stelle entspricht, dem das positive Relationssymbol R_j zugeordnet ist. Handelt es sich hingegen um ein negatives Literal der Form $\neg R_j(t_{j_1},...,t_{j_{l(j)}})$, so wird die Flussrelation FR um die Kante (tr_n,st_m) erweitert, wobei $st_m \in ST_{NEG}$ mit $st_m=AN_{RS}(\underline{R}_j)$ der negativen Stelle entspricht, dem das negative Relationssymbol $\underline{R}_j$ zugeordnet ist.

Die Transformation einer Inferenzregel $F \in INF_{SIG_{OS}}$ in ein Ontologie-Teilnetz TN erfolgt nach folgendem Verfahren:

FORALL $F \in INF_{SIG_{OS}}$ AND $F = (\forall x_1,...,x_k: lit_1 \wedge .... \wedge lit_n \rightarrow lit_{n+1})$ DO

1.) *Initialisierung*

$AN_{TR}(trans_{INF}(F)) := w$

1) Die konjunktive Erweiterung wird dann verwendet, wenn in diesem Iterationsschritt die Transitionsannotation $AN_{TR}(tr_n)$ nicht mehr mit der tautologischen Formel w aus der Initialisierung übereinstimmt.

2) Wenn das Konklusionsliteral lit_{n+1} mit $lit_{n+1}=R_{n+1}(t_{n+1_1},...,t_{n+1_{l(n+1)}})$ aufgrund $R_{n+1} \in RS_{ST}$ mit einem statischen Relationssymbol zusammengesetzt ist, muss die Inferenzregel als Integritätsregel behandelt werden und entsprechend in eine Integritätstransition transformiert werden. Zwar wird dadurch die dynamische Erschließung von Fakten ausgeschlossen, allerdings können Markierungen aufgedeckt werden, die bezüglich der Integritätsregel unzulässig sind.

2.) *Transformation*

FORALL j=1 TO n DO

CASE$_1$ $lit_j = R_j(t_{j_1},\ldots,t_{j_{l(j)}})$

CASE$_{1_1}$ $R_j \in RS_{DY}$

$IR := IR \cup \{(AN_{RS}(R_j), trans_{INF}(F))\}$

$AN_{IR}(AN_{RS}(R_j), trans_{INF}(F)) := AN_{IR}(AN_{RS}(R_j), trans_{INF}(F)) \oplus \langle t_{j_1},\ldots,t_{j_{l(j)}} \rangle$

END CASE$_{1_1}$

CASE$_{1_2}$ $R_j \in RS_{ST}$

IF $AN_{TR}(trans_{INF}(F)) = w$

THEN

$AN_{TR}(trans_{INF}(F)) := R_j(t_{j_1},\ldots,t_{j_{l(i)}})$

ELSE

$AN_{TR}(trans_{INF}(F)) := AN_{TR}(trans_{INF}(F)) \wedge R_j(t_{j_1},\ldots,t_{j_{l(i)}})$

END IF

END CASE$_{1_2}$

END CASE$_1$

CASE$_2$ $lit_j = \neg R_j(t_{j_1},\ldots,t_{j_{l(j)}})$

CASE$_{2_1}$ $R_j \in RS_{DY}$

$IR := IR \cup \{(AN_{RS}(\underline{R}_j), trans_{INF}(F))\}$

$AN_{IR}(AN_{RS}(\underline{R}_j), trans_{INF}(F)) := AN_{IR}(AN_{RS}(\underline{R}_j), trans_{INF}(F)) \oplus \langle t_{j_1},\ldots,t_{j_{l(j)}} \rangle$

END CASE$_{2_1}$

CASE$_{2_2}$ $R_j \in RS_{ST}$

IF $AN_{TR}(trans_{INF}(F)) = w$

THEN

$AN_{TR}(trans_{INF}(F) := \neg R_j(t_{j_1},\ldots,t_{j_{l(i)}})$

ELSE

$AN_{TR}(trans_{INF}(F)) := AN_{TR}(trans_{INF}(F)) \wedge \neg R_j(t_{j_1},\ldots,t_{j_{l(i)}})$

END IF

END CASE$_{2_2}$

END CASE$_2$

END DO

IF $lit_{n+1} = R_{n+1}(t_{n+1_1},\ldots,t_{n+1_{l(n+1)}})$

THEN

$FR := FR \cup \{(trans_{INF}(F), AN_{RS}(R_{n+1}))\}$

$AN_{FR}(trans_{INF}(F), AN_{RS}(R_{n+1})) := AN_{FR}(trans_{INF}(F), AN_{RS}(R_{n+1})) \oplus \langle t_{n+1_1},\ldots,t_{n+1_{l(n+1)}} \rangle$

END IF

IF $lit_{n+1}=\neg R_{n+1}(t_{n+1_1},\ldots,t_{n+1_{l(n+1)}})$

THEN

$FR := FR \cup \{(trans_{INF}(F), AN_{RS}(\underline{R}_{n+1}))\}$

$AN_{FR}(trans_{INF}(F), AN_{RS}(\underline{R}_{n+1})) := AN_{FR}(trans_{INF}(F), AN_{RS}(\underline{R}_{n+1})) \oplus <t_{n+1_1},\ldots,t_{n+1_{l(n+1)}}>$

END IF

Mit Hilfe des algorithmischen Transformationsverfahrens kann beispielsweise die Inferenzregel

$$\forall x_1,x_2: (R_1(x_1,x_2) \wedge \neg R_2(x_1,x_2) \wedge \neg Equal(x_1,x_2)) \rightarrow R_3(x_1,x_2)$$

mit den dynamischen Relationssymbolen R_1, R_2 und R_3 und dem statischen Relationssymbol *equal* entsprechend Abbildung 24 in ein Ontologie-Teilnetz transformiert werden.

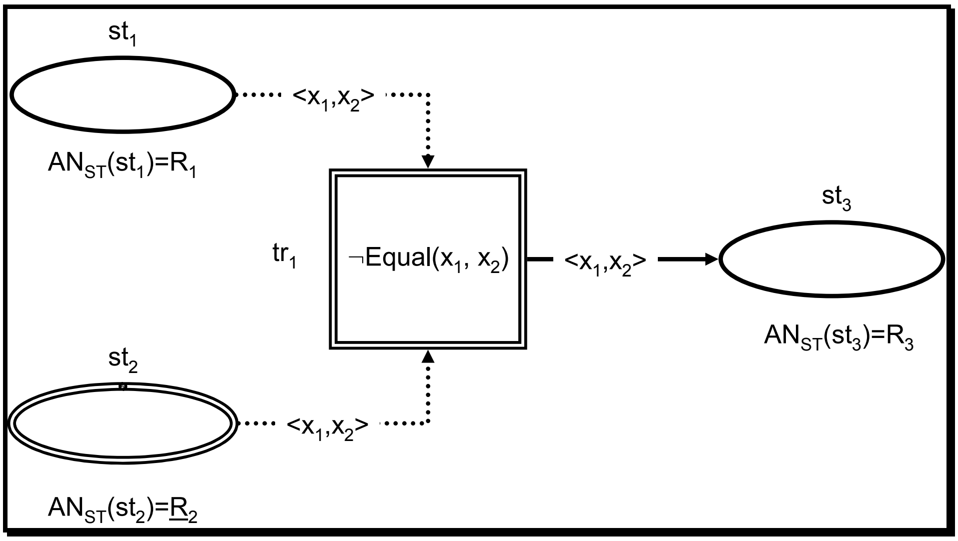

Abbildung 24: Transformierte Inferenzregel

Inferenztransitionen, die aus der algorithmischen Transformation von Inferenzregeln hervorgehen, zeichnen sich durch zwei wesentliche Eigenschaften aus. Die erste Eigenschaft besteht darin, dass Inferenztransitionen als Eingangskanten nur Informationskanten aufweisen können. Durch das Schalten von Inferenztransitionen gilt es nämlich von einer Referenzmarkierung M_r zu einer solchen Folgemarkierung M_f überzugehen, bei der keine bestehenden Fakten aufgehoben werden, sondern nur neue Fakten generiert werden. Der Ausschluss der Faktenentfernung wird dadurch gewährleistet, dass als Eingangskanten von Inferenztransitionen nur Informationskanten in Frage kommen. Denn Informationskanten unterscheiden sich von Flusskanten dadurch, dass Informationskanten bei einem Schaltakt keine Markierungsvariation bewirken. In ihrer zweiten Eigenschaft setzen Inferenztransitionen an dem Potenzial zur Faktenvermehrung durch Flusskanten an. Bei jedem Schaltakt einer Inferenztransition wird gemäß der Annotation der Ausgangskante mindestens ein ontologisches Termtupel auf der Ausgangsstelle abgelegt, was dem Hinzufügen eines Faktums zur Faktenbasis entspricht.

In ihrer Gesamtheit stellen diese beiden Eigenschaften von Inferenztransitionen eine bemerkenswerte Korrespondenz zu der bereits angesprochenen Monotonieprämisse der prädikatenlogischen Wissensrepräsentation her. Der prädikatenlogischen Monotonieprämisse folgend darf durch das Hinzufügen von einer gültigen Formel F zu einer gültigen Formelmenge FM die Gültigkeit von FM nicht verletzt werden.[1)] Die Korrespondenz von Inferenztransitionen mit der Monotonieprämisse der Prädikatenlogik spiegelt sich in den unterschiedlichen Kantenarten wieder, durch die Inferenztransitionen mit ihrer Nachbarschaft verbunden sein können. Da die Stelle st_m im Nachbereich $NB_{TR}(tr_n)$ einer Inferenztransition tr_n mittels Flusskanten mit tr_n verbunden sein müssen, kann es durch den Schaltakt von tr_n dazu kommen, dass die Folgemarkierung M_f der Stelle st_m eine Multimenge von Marken zuordnet, in der auch Marken enthalten sind, die in der Referenzmarkierung M_r nicht enthalten gewesen sind. Dadurch käme es zu einer *Vergrößerung* der Faktenmenge. Hingegen kann es durch das Schalten einer Inferenztransition nicht zu einer *Verkleinerung* der Faktenmenge kommen, da der Markenentzug durch einen Schaltakt nur für Stellen im Vorbereich $VB_{TR}(tr_n)$ von tr_n in Frage kommen kann. Bei den Eingangskanten von Inferenztransitionen kann es sich jedoch nur um Informationskanten handeln, die zwar für die Aktivierungsbedingungen Relevanz haben, allerdings nicht für die Schaltwirkung.

1) Zur prädikatenlogische Monotonie vgl. ANTONIOU (1996), S. 166 f.

4.2.2.1.2.2 Integritätstransitionen

Analog zur Transformation von Inferenzregeln können auch Integritätsregeln in ein Teilnetz TN des Ontologie-Netzes ON transformiert werden. Teilnetze, die aus der Transformation von Integritätstransitionen hervorgehen, weisen jedoch als einzige Transition jeweils eine *regelspezifische*[1)] *Integritätstransition* auf.

Integritätstransitionen haben mit Inferenztransitionen gemeinsam, dass sie als Eingangskanten nur Informationskanten aufweisen. Obwohl Schaltakte von Integritätstransitionen nicht von Relevanz sind, da die Erfülltheit der Schaltvoraussetzungen von Integritätstransitionen auf eine Unzulässigkeit der betrachteten Markierung hinweisen, werden Integritätstransitionen mit ihren inzidenten Stellen nur mit Informationskanten verbunden, durch die grundsätzlich keine Markierungsvariation bewirkt werden kann.[2)] Darüber hinausgehende Verknüpfungen von Inferenztransitionen mit Stellen sind nicht erlaubt.

Integritätsregeln dienen in einer Ontologie $SPEZ_{OS}$ dazu, die Zulässigkeit von Formeln zu überprüfen. Dabei ist eine Formelmenge $FM \in FORM_{SIG_{OS}}$ genau dann bezüglich einer Integritätsregel $F \in INT_{SIG_{OS}}$ zulässig, wenn die Formelmenge $FM \cup F$ nicht widersprüchlich ist. Die Transformation von Integritätsregeln in ein Ontologie-Teilnetz setzt genau an dieser Funktionalität von Integritätsregeln an. Die Transformation verläuft nämlich derart, dass die regelspezifische Integritätstransition $trans_{INT}(F)=tr_n$ zu einer Integritätsregel $F \in INT_{SIG_{OS}}$ genau dann unter einer Markierung M_z aktiviert ist, wenn M_z

---

1) Der Zusatz „regelspezifische" wird in Zusammenhang mit Integritätsregeln dann verwendet, wenn aus dem Argumentationskontext nicht hervorgeht, dass es sich um eine Integritätstransition tr_n handelt, die einer Integritätsregel $F \in INT_{SIG_{OS}}$ mittels $trans_{INT}(F)=tr_n$ zugeordnet ist. Im weiteren Verlauf werden auch solche Integritätstransitionen vorgestellt, die keinen Integritätsregeln aus der Ontologie $SPEZ_{OS}$ zugeordnet sind.

2) Grundsätzlich bestünde auch für Integritätstransitionen die Möglichkeit, als Eingangskanten Flusskanten zuzulassen. Dies hätte sogar den Vorteil, dass, wenn Schaltakte für Integritätstransitionen zugelassen würden, die integritätsverletzenden Marken von den inzidenten Stellen abgezogen würden. Dadurch könnte die Integritätsverletzung aufgehoben werden. Allerdings wird diese Option für Ontologie-Netze aus zwei Gründen nicht in Betracht gezogen. Erstens wird durch die alleinige Zulässigkeit adjazenter Informationskanten zu Integritätstransitionen die Ähnlichkeit zu der Vorgehensweise bei Inferenztransitionen beibehalten. Denn auch für Inferenztransitionen kommen als Eingangskanten nur Informationskanten in Betracht. Zweitens würden durch den Schaltakt einer Integritätsverletzung *alle* Marken von den Vorbereichsstellen entfernt werden, obwohl vielleicht nur eine klauselbezogene Stellenmarkierung für die Integritätsverletzung verantwortlich ist. Daher werden für Ontologie-Netze grundsätzlich Schaltakte von Integritätstransitionen nicht berücksichtigt.

im Widerspruch zu F ist. Somit entspricht die Aktiviertheit einer Integritätstransition tr_n der Unzulässigkeit[1)] einer Markierung.[2)]

Auch für die algorithmische Transformierbarkeit einer Integritätsregel in ein Ontologie-Teilnetz TN muss – analog zu der Transformation von Inferenzregeln – eine regelspezifische Integritätstransition $trans_{INT}(F)=tr_n$ vorliegen. Bevor die Transformation durchgeführt werden kann, muss auch hierbei die Annotation $AN_{TR}(tr_n)$ der Integritätstransition tr_n mit der tautologischen Formel w initialisiert werden. Auch die Transformation der Antezedenz-Literale von Integritätsregeln verläuft identisch zu der Transformation der Antezedenz-Literale von Inferenzregeln. Es werden nämlich auch hierbei alle Stellen, die den dynamischen Relationssymbolen aus den Antezedenz-Literalen von F zugeordnet sind, über Informationskanten mit tr_n verbunden und entsprechend die Kanten annotiert. Die Unterscheidung zwischen positiven und negativen Stellen erfolgt analog.

Hinsichtlich der Behandlung von Konklusions-Literalen unterscheidet sich die Transformation von Integritätsregeln wesentlich von der Transformation von Inferenzregeln. Erstens haben Integritätstransitionen keine Nachbereichsstellen, da sie lediglich zur Überprüfung der Zulässigkeit von Markierungen verwendet werden. Entsprechend weisen sie auch keine adjazenten Flusskanten auf. Zweitens wird im Fall von Integritätsregeln an der Kontradiktion des Konklusions-Literals angesetzt. Denn eine regelspezifische Integritätstransition $trans_{INT}(F)=tr_n$ hat genau dann aktiviert zu sein, wenn die Markierung M_z im Widerspruch zu F steht. Entsprechend müssen die Bedingungen für die Aktiviertheit der Integritätstransition so formuliert werden, dass sie kontradiktorisch zu F sind. Die Kontradiktion zu einer Integritätsregel

$$F = \forall x_1,...,x_k\colon lit_1 \wedge .... \wedge lit_n \rightarrow lit_{n+1}$$

lautet

$$\hat{F} = \forall x_1,...,x_k\colon lit_1 \wedge .... \wedge lit_n \wedge \neg lit_{n+1}$$

Handelt es sich bei dem Konklusions-Literal lit_{n+1} der Integritätsregel um ein positives Literal der Form $R_{n+1}(t_{n+1_1},...,t_{n+1_{l(n+1)}})$, das mit einem dynamischen Relationssymbol $R_{n+1} \in RS_{DY}$ konstruiert ist, so wird die Integritätstransition tr_n mittels einer Informationskante $(st_m,tr_n) \in IR$ mit der Stelle st_m verbunden, die der Negation $\underline{R}_{n+1}$ des dynami-

1) Die Unzulässigkeit einer Markierung M_z bezüglich einer Integritätstransition $tr_n \in TR_{INT}$ ist von der Unzulässigkeit von M_z bezüglich einer der Integritätsbedingungen IB_{ON_3} und IB_{ON_4} zu unterscheiden. Die Integritätsbedingungen IB_{ON_3} und IB_{ON_4} sind auf *alle* stellenbezogenen Markierung eines Ontologie-Netzes in dem Zustand z bezogen. Dies wird sowohl bei IB_{ON_3} als auch bei IB_{ON_4} durch die Allquantifizierung über der Menge ST aller Stellen deutlich. Die Unzulässigkeit einer Markierung bezüglich einer Integritätstransition tr_n kann hingegen nur *lokal* bestimmt werden. Mit einer Integritätstransition tr_n können nämlich nur solche unzulässigen stellenbezogenen Markierungen kenntlich gemacht werden, die sich auf Stellen in der Nachbarschaft $NA_{TR}(tr_n)$ von tr_n beziehen.

2) Es wird hier vereinfacht von der Unzulässigkeit eine Markierung M_z bezüglich einer Integritätsregel gesprochen. Eigentlich ist die Unzulässigkeit jedoch nicht auf die Markierung M_z bezogen, sondern auf die Formelmenge FM, deren Gültigkeit durch die Markierung M_z ausgedrückt wird.

schen Relationssymbols R_{n+1} zugeordnet ist ($AN_{RS}(\underline{R}_{n+1})=st_m$). Handelt es sich hingegen bei R_{n+1} um ein statisches Relationssymbol, so ist die Transitionsannotation $AN_{TR}(tr_n)$ entweder das negative Literal $\neg R_{n+1}(t_{n+1_1},...,t_{n+1_{l(n+1)}})$ oder $AN_{TR}(tr_n)$ wird konjunktiv mit dem Literal verknüpft. Analog wird – wenn es sich bei dem Konklusions-Literal lit_{n+1} um ein negatives Literal der Form $\neg R_{n+1}(t_{n+1_1},...,t_{n+1_{l(n+1)}})$ handelt – die Stelle st_m mit $AN_{RS}(R_{n+1})=st_m$ mittels einer Informationskante (st_m,tr_n) mit der Integritätstransition tr_n verbunden, wenn R_{n+1} ein dynamisches Relationssymbol ist. Wenn jedoch R_{n+1} ein statisches Relationssymbol ist, ist die Annotation $AN_{TR}(tr_n)$ der Integritätstransition tr_n entweder das positive Literal $R_{n+1}(t_{n+1_1},...,t_{n+1_{l(n+1)}})$ oder $AN_{TR}(tr_n)$ wird konjunktiv mit dem Literal verknüpft.

Die algorithmische Transformation einer Integritätsregel $F \in INT_{SIG_{OS}}$ in ein Ontologie-Teilnetz TN erfolgt nach folgendem Schema:

FORALL $F \in INT_{SIG_{OS}}$ AND $F = (\forall x_1,...,x_k: (lit_1 \wedge .... \wedge lit_n) \rightarrow lit_{n+1})$ DO

1.) *Initialisierung*

$AN_{TR}(trans_{INT}(F)) = w$

2.) *Transformation*

FORALL j=1 TO n DO

  $CASE_1\ lit_j = R_j(t_{j1},\ldots,t_{jl(j)})$

    $CASE_{1_1}\ R_j \in RS_{DY}$

      $IR := IR \cup \{(AN_{RS}(R_j), trans_{INT}(F))\}$

      $AN_{IR}(AN_{RS}(R_j), trans_{INT}(F)) := AN_{IR}(AN_{RS}(R_j), trans_{INT}(F)) \oplus \langle t_{j1},\ldots,t_{jl(j)} \rangle$

    $END\ CASE_{1_1}$

    $CASE_{1_2}\ R_j \in RS_{ST}$

      $IF\ AN_{TR}(trans_{INT}(F)) = w$

        THEN

          $AN_{TR}(trans_{INT}(F)) \quad := \quad R_j(t_{j1},\ldots,t_{jl(i)})$

        ELSE

          $AN_{TR}(trans_{INT}(F)) \quad := \quad AN_{TR}(trans_{INT}(F)) \wedge R_j(t_{j1},\ldots,t_{jl(j)})$

      END IF

    $END\ CASE_{1_2}$

  $END\ CASE_1$

  $CASE_2\ lit_j = \neg R_j(t_{j1},\ldots,t_{jl(i)})$

    $CASE_{2_1}\ R_j \in RS_{DY}$

      $IR := IR \cup \{(AN_{RS}(\underline{R}_j), trans_{INT}(F))\}$

      $AN_{IR}(AN_{RS}(\underline{R}_j), trans_{INT}(F)) := AN_{IR}(AN_{RS}(\underline{R}_j), trans_{INT}(F)) \oplus \langle t_{j1},\ldots,t_{jl(j)} \rangle$

    $END\ CASE_{2_1}$

    $CASE_{2_2}\ R_j \in RS_{ST}$

      $IF\ AN_{TR}(trans_{INT}(F)) = w$

        THEN

          $AN_{TR}(trans_{INT}(F) \quad := \quad \neg R_j(t_{j1},\ldots,t_{jl(j)})$

        ELSE

          $AN_{TR}(trans_{INT}(F)) \quad := \quad AN_{TR}(trans_{INT}(F)) \wedge \neg R_j(t_{j1},\ldots,t_{jl(i)})$

    $END\ CASE_{2_2}$

  $END\ CASE_2$

END DO

CASE$_3$ $lit_{n+1}=R_{n+1}(t_{n+1_1},...,t_{n+1_{l(n+1)}})$

  CASE$_{3_1}$ $R_{n+1}\in RS_{DY}$

    $IR := IR \cup \{(AN_{RS}(\underline{R}_{n+1})),trans_{INT}(F)\}$

    $AN_{IR}(AN_{RS}(\underline{R}_{n+1}),trans_{INT}(F)) := AN_{IR}(AN_{RS}(\underline{R}_{n+1})),trans_{INT}(F)) \oplus <t_{n+1_1},...,t_{n+1_{l(n+1)}}>)$

  END CASE$_{3_1}$

  CASE$_{3_2}$ $R_{n+1}\in RS_{DY}$

    IF $AN_{TR}(trans_{INT}(F)) = w$

      THEN

        $AN_{TR}(trans_{INT}(F)) \quad := \quad \neg R_{n+1}(t_{n+1_1},...,t_{n+1_{l(n+1)}})$

      ELSE

        $AN_{TR}(trans_{INT}(F)) \quad :=AN_{TR}(trans_{INT}(F)) \wedge \neg R_{n+1}(t_{n+1_1},...,t_{n+1_{l(n+1)}})$

    END IF

  END CASE$_{3_2}$

CASE$_4$ $lit_{n+1}=\neg R_{n+1}(t_{n+1_1},...,t_{n+1_{l(n+1)}})$

  CASE$_{4_1}$ $R_{n+1}\in RS_{DY}$

    $IR := IR \cup \{(AN_{RS}(R_{n+1})),trans_{INT}(F)\}$

    $AN_{IR}(AN_{RS}(R_{n+1}),trans_{INT}(F)) := AN_{IR}(AN_{RS}(R_{n+1}),trans_{INT}(F)) \oplus <t_{n+1_1},...,t_{n+1_{l(n+1)}}>$

  END CASE$_{4_1}$

  CASE$_{4_2}$ $R_{n+1}\in RS_{ST}$

    IF $AN_{TR}(trans_{INT}(F)) = w$

      THEN

        $AN_{TR}(trans_{INT}(F)) \quad := \quad R_{n+1}(t_{n+1_1},...,t_{n+1_{l(i)}})$

      ELSE

        $AN_{TR}(trans_{INT}(F)) \quad := AN_{TR}(trans_{INT}(F) \wedge R_{n+1}(t_{n+1_1},...,t_{n+1_{l(n+1)}})$

    END IF

  END CASE$_{4_2}$

END CASE$_4$

Die Formel die in Abschnitt 4.2.2.1.2.1 als Inferenzregel vorgestellt und in eine Inferenztransition transformiert wurde, kann nun als Integritätsregel aufgefasst und in eine Integritätstransition transformiert werden. Hierdurch wird die Unterscheidung zwischen den beiden Regelarten auf *pragmatischer* Ebene deutlich.

Die Integritätsregel

$$\forall x_1,x_2: (R_1(x_1,x_2) \wedge \neg R_2(x_1,x_2) \wedge \neg Equal(x_1,x_2)) \rightarrow R_3(x_1,x_2)$$

mit den dynamischen Relationssymbolen R_1, R_2 und R_3 und dem statischen Relationssymbol Equal kann entsprechend der Abbildung 25 transformiert werden.

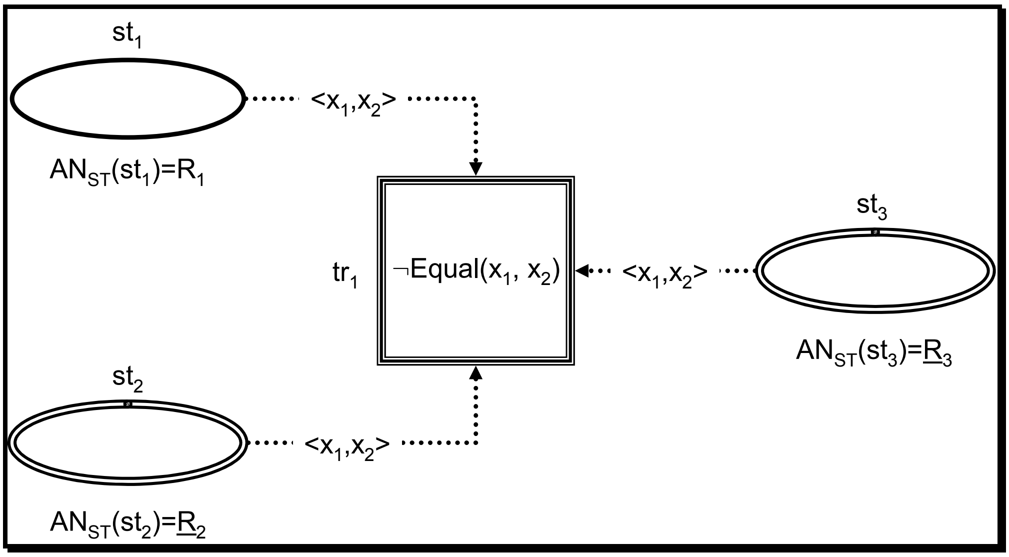

Abbildung 25: Transformierte Integritätsregel

Mit Integritätstransitionen können *Business-Rules*, die in allen Zuständen eines ontologiegestützten Modells zu gelten haben, auf anschauliche Weise ausgedrückt werden.[1)] Hierbei können auch Business Rules ausgedrückt werden, die sich aus dem Zusammenwirken von Inferenz- und Integritätstranstionen ergeben.[2)] Mit der Inferenzregel

$\forall MA_1,MA_2,MA_3$: $(Vorgesetzter\_von(MA_1,MA_2) \wedge Vorgesetzter\_von(MA_2,MA_3))$
$\rightarrow Vorgesetzter\_von\ (MA_1,MA_3)$

wird die Transitivität der Vorgesetzten-Relation ausgedrückt. Mit der Integritätsregel

$\forall MA_1,MA_2,AA$: $Vorgesetzter\_von(MA_2,MA_1) \rightarrow \neg Arbeitsanweisung\_an(MA_1,AA,MA_2)$

wird der Ausschluss von Arbeitsanweisungen von Mitarbeitern an ihre Vorgesetzten ausgedrückt. Wenn ein Mitarbeiter MA_2 Vorgesetzter eines Mitarbeiters MA_1 ist, darf keine Beziehung existieren, der zufolge MA_1 an MA_2 eine Arbeitsanweisung AA gibt.

Die beiden Regeln können entsprechend Abbildung 26 zu einem Ontologie-Teilnetz transformiert werden.

1) Vgl. OBERWEIS (1996), S. 197 ff.

2) Bezüglich des Zusammenwirkens von Inferenz- und Integritätstransitionen sind in Ontologie-Netzen *Aktivierungsprioritäten* zu beachten, denen zu Folge keine Inferenztransition aktiviert werden darf, so lange mindestens eine Integritätstransition aktiviert ist. Diese Aktivierungsprioritäten werden im nächsten Abschnitt thematisiert.

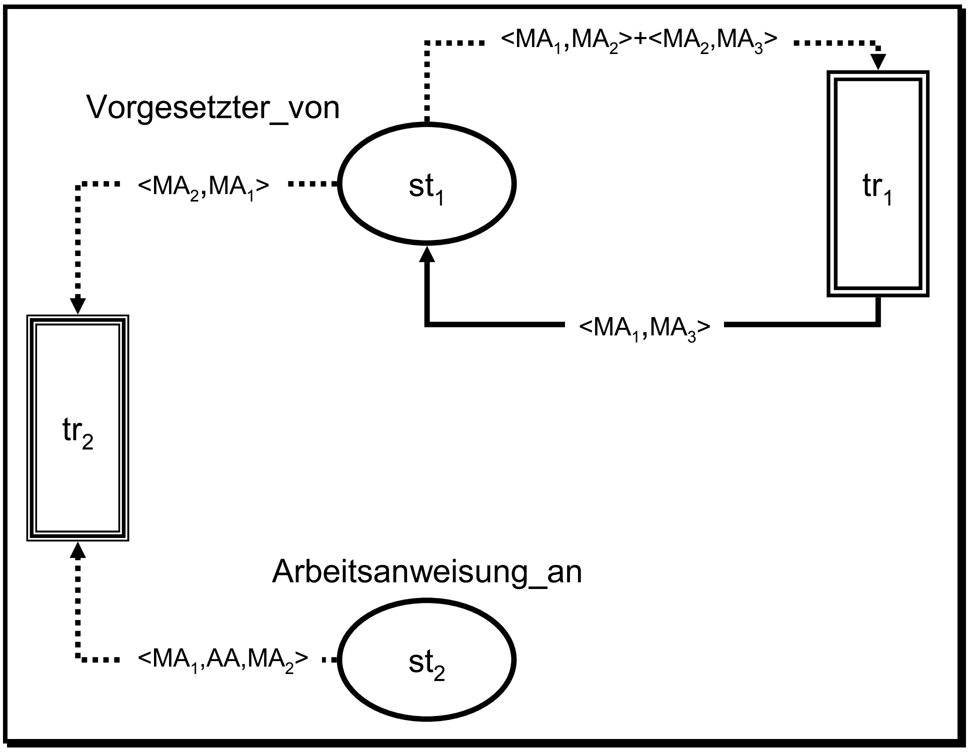

Abbildung 26: Zusammenwirken von Inferenz- und Integritätstransitionen

Mit Hilfe der Inferenztransition tr_1 werden sämtliche Fakten bezüglich der Vorgesetzten-Beziehung erschlossen. Mit der Integritätstransition tr_2 werden hingegen unzulässige Markierungen verhindert, die unzulässigen Faktenmengen entsprechen würden. Durch die Gesamtheit des Ontologie-Teilnetzes aus Abbildung 26 werden zwei Business Rules berücksichtigt. Einerseits gilt in der betrachteten Domäne, dass die Vorgesetzten-Beziehung transitiv ist. Andererseits dürfen keine Arbeitsanweisungen von Mitarbeitern an ihre Vorgesetzten verschickt werden. Die beiden Business-Rules überschneiden sich in dem Punkt, in dem beide auf die Vorgesetzten-Relation Bezug nehmen. In der graphischen Darstellung findet sich das in den beiden Informationskanten (st_1,tr_1) und (st_1,tr_2) wieder, anhand derer die Transitionen mit der Stelle st_1 verbunden sind.

Über die Integritätstransitionen hinaus, die aus der Transformation von Integritätsregeln hervorgehen, können in einem Ontologie-Netz ON noch weitere Integritätstransitionen vorhanden sein, denen in der zugrunde gelegten Ontologie $SPEZ_{OS}$ keine Integritätsregel $F \in INT_{SIG_{OS}}$ entspricht. Hierzu kommen solche Integritätstransitionen in Frage, mit denen die *Konsistenz* der Stellenmarkierung M_z in jedem Zustand $z \in \mathbb{N}$ bewahrt wird. Die Konsistenz der Stellenmarkierung M_z ist in einem Ontologie-Netz ON immer dann verletzt, wenn im Zustand für mindestens ein Relationssymbol $R_j \in RS_{DY}$

$$M_z(AN_{RS}(R_j)) \cap M_z(AN_{RS}(\underline{R}_j)) \neq \varnothing_M.$$

gilt. Demnach darf ein Termtupel $<t_1,...,t_n>$ nicht gleichzeitig in der Markierung von zwei Stellen $st_1 \in ST_{POS}$ und $st_2 \in ST_{NEG}$ mit $AN_{RS}(R_j)=st_1$ bzw. $AN_{RS}(\underline{R}_j)=st_2$ enthalten sein. In diesem Fall würden sich die extensionalen Interpretationen des positiven Relationssymbols R_j und des dazu komplementären negativen Relationssymbols $\underline{R}_j$ „überlappen“. Eine solche Markierung würde einer widersprüchlichen Faktenmenge entsprechen, weil sie sowohl das Faktum $R_j(t_1,...,t_n)$ als auch das hierzu kontradiktorische Faktum $\neg R_j(t_1,...,t_n) \equiv \underline{R}_j(t_1,...,t_n)$ enthielte.

Um die Konsistenz der zustandsbezogenen Markierungen zu gewährleisten, wird in Ontologie-Netzen zu jeder negativen Stelle $st_1 \in ST_{NEG}$ mit $AN_{ST}(st_1)=\underline{R}_1$ eine Integritätstransition $tr_n \in TR_{INT}$ spezifiziert, die in ihrem Vorbereich $VB_{TR}(tr_n)$ genau die zwei Stellen st_1 und $st_2 \in ST_{POS}$ mit $AN_{ST}(st_2)=R_j$ aufweist. Die Kanten $(st_1,tr_n),(st_2,tr_n) \in IR$ werden jeweils mit dem gleichen ontologischen Termtupel $<x_1,...,x_n>$, das nur mit Hilfe von Variablen konstruiert ist, annotiert. Abbildung 27 enthält ein Beispiel für eine Integritätstransition zur Bewahrung der Konsistenz in einem Ontologie-Netz ON.

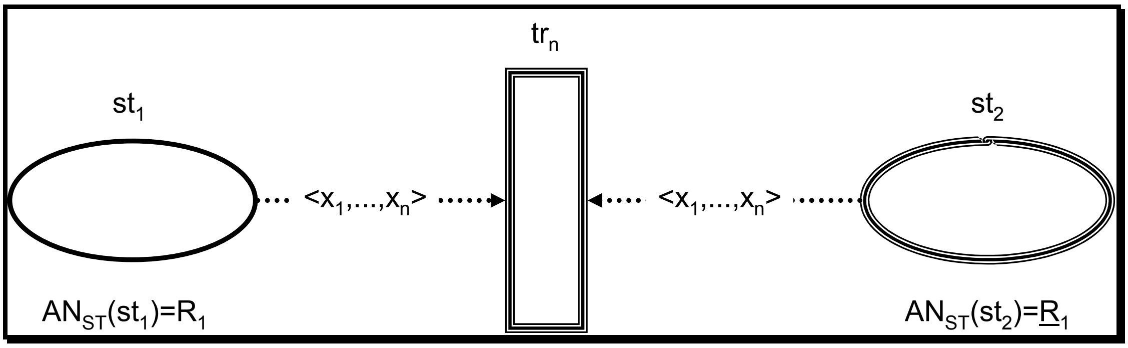

Abbildung 27: Integritätstransition zur Bewahrung der Markierungskonsistenz

4.2.2.2 Dynamische Struktur von Ontologie-Netzen

4.2.2.2.1 Schaltregeln in Ontologie-Netzen

4.2.2.2.1.1 Schaltregel für einzelne Transitionen in Ontologie-Netzen

Die statische Struktur eines Ontologie-Netzes umfasst neben der Netztopologie eine Ontologie $SPEZ_{OS}$, die Transitionszuordnungsfunktionen $trans_{INF}$ und $trans_{INT}$, die statische SIG_{ST}-Struktur $A_{SIG_{ST}}$, die Annotationsfunktionen $(AN_{ST},AN_{TR},AN_{FR},AN_{IR})$, die Kapazitätsfunktion KP und die Anfangsmarkierungsfunktion M_0. Insbesondere durch die letzte Komponente wird deutlich, dass in der statischen Struktur eines Ontologie-Netzes ON lediglich *ein* Zustand – nämlich der Initialzustand – von ON abgebildet wird. Die Anfangsmarkierung M_0 wird – analog zu der initialen Markierung von S/T-Netzen – als exogen vorgegeben angenommen. Sämtliche (Folge-)Markierungen M_z mit $z=1,...,Z$ und $Z \in \mathbb{N}$ ergeben sich endogen aus der *dynamischen Struktur* des Ontologie-Netzes.

Die dynamische Struktur von Ontologie-Netzen wird in Form einer *Schaltregel* operationalisiert, die sich aus zwei Komponenten zusammensetzt. Die erste Komponente der Schaltregel zu einem Ontologie-Netz ON entspricht den *Voraussetzungen*, die erfüllt sein müssen, damit eine Transition $tr_n \in TR$ schalten kann. Die Schaltvoraussetzungen für Transitionen stimmen mit ihren *Aktivierungsbedingungen* überein. Die zweite Komponente der dynamischen Struktur ist die *Wirkung*, die das Schalten einer Transition $tr_n \in TR$ auf das Ontologie-Netz ON hat. Sie wird in Form einer *Schaltfunktion* operationalisiert.

Um die Aktivierung einer Transition tr_n unter einer Variablensubstitution θ und einer zulässigen Markierung M_r auszudrücken, wird die Relation

$$AKT_{ON} \subseteq (TR \times 2^{\theta} \times ZMF_{ON})$$

verwendet. Mit dem Ausdruck $AKT_{ON}(tr_n,\theta,M_r)$ wird angegeben, dass die Transition $tr_n \in TR$ unter der Familie $\theta \in 2^{\theta}$ konzeptspezifischer Variablensubstitutionsfunktionen[1)] und der Markierung $M_r \in ZMF_{ON}$ aktiviert ist. Als Variablensubstitution θ kommen für die Aktivierung einer Transition tr_n nur Substitutionen in betracht, denen zufolge alle Variablen in einem Ausdruck durch Grundterme ersetzt werden. Diese besonderen Variablensubstitutionen werden im Folgenden auch als Grundsubstitutionen bezeichnet.

Bei der Untersuchung, ob eine Transition tr_n unter einer Grundsubstitution θ und einer Markierung M_r aktiviert ist, muss zunächst festgestellt werden, ob eine deklarative oder prozedurale Transition vorliegt. Liegt eine deklarative Transition vor, so kann es sich nur um eine Inferenz- oder Integritätstransition handeln. Die Aktiviertheit von Integritätstransitionen weist grundsätzlich auf die Unzulässigkeit der Markierung M_r hin. In diesem Fall werden Eingriffe in die Markierung des Ontologie-Netzes notwendig, um die Konsistenz der Markierung wieder herzustellen. Liegt hingegen eine aktivierte Inferenztransition vor, so lässt sich aus der Markierung M_r mindestens eine weitere Markierung M_f ableiten, die zusätzlich zu den Formeln, deren Gültigkeit aus M_r hervorgeht, weitere gültige Formeln umfasst.

Um die verschiedenen Transitionsarten bei der Definition ihrer Schaltvoraussetzungen berücksichtigen zu können, werden in ihren Aktivierungsbedingungen *Aktivierungsprioritäten* eingeräumt.[2)] Die Abbildung 28 gibt die Anordnung der verschiedenen Transitionsarten entsprechend ihren Aktivierungsprioritäten wieder.

---

1) Für die Bezeichnung „Familie konzeptspezifischer Variablensubstitutionsfunktionen" wurde in Abschnitt 3.1.4.3.1 auch die verkürzte Bezeichnung „Variablensubstitution" zugelassen.

2) Die Aktivierungspriorität ist von der *Schaltpriorität* abzugrenzen (zu Schaltprioritäten vgl. STARKE (1990), S. 103 f.). Im Fall von Petri-Netzen mit Schaltprioritäten können die Transitionen nur entsprechend der Reihenfolge schalten, die durch eine partielle Ordnung auf der Menge TR aller Transitionen vorgegeben ist. Demnach kann eine Transition tr_1 mit einer geringeren Schaltpriorität als eine weitere Transition tr_2 gleichzeitig mit dieser aktiviert sein. Sie darf nur nicht vor tr_2 schalten. Dadurch werden für Transitionen neben den Aktivierungsbedingungen weitere Schaltvoraussetzungen geltend gemacht. In Ontologie-Netzen wird hingegen durch Aktivierungsprioritäten die Eingrenzung der Schaltvoraussetzungen auf die Aktivierungsbedingungen beibehalten.

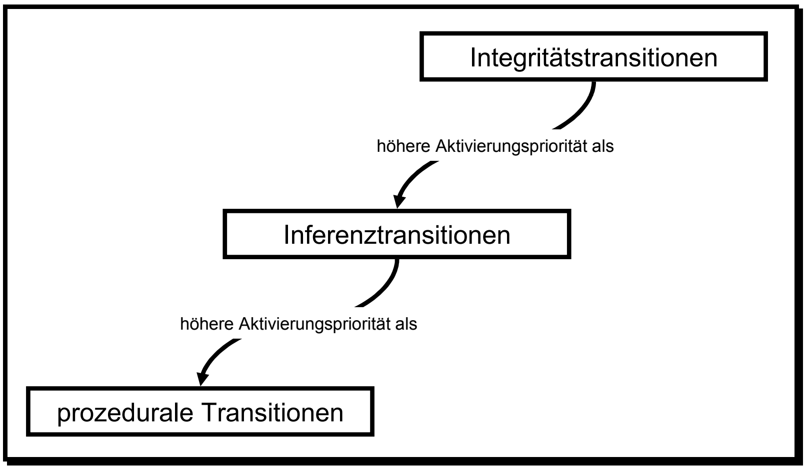

Abbildung 28: Aktivierungsprioritäten in Ontologie-Netzen

Die höchste Aktivierungspriorität wird Integritätstransitionen eingeräumt. Inferenztransitionen wird im Vergleich zu Integritätstransitionen eine niedrigere Aktivierungspriorität eingeräumt. Die Aktivierungsprioritäten prozeduraler Transitionen sind schließlich niedriger als die Aktivierungsprioritäten aller deklarativen Transitionen.

Für eine Integritätstransition $tr_n \in TR_{INT}$ gilt unter einer Grundsubstitution θ und einer Markierung M_r genau dann $AKT_{ON}(tr_n,\theta,M_r)$, wenn

(1.) $\forall st_m \in VB_{TR}(tr_n)$: $(AN_{IR}(st_m,tr_n)[\theta]) \subseteq M_r(st_m)$ und

(2.) $A_{SIG_{ST}} \models (AN_{TR}(tr_n)[\theta])$

gelten. Der *ersten* Aktivierungsbedingung für Integritätstransitionen zufolge muss jede Stelle st_m aus dem Vorbereich $VB_{TR}(tr_n)$ von tr_n eine Markierung $M_r(st_m)$ aufweisen, die mindestens den Ausdruck umfasst, der aus der Ersetzung der Informationskantenannotation $AN_{IR}(st_m,tr_n)$ entsprechend der Grundsubstitution θ hervorgeht. Mit dieser Aktivierungsbedingung werden die *dynamischen* Aspekte der Schaltvoraussetzung überprüft. Die *statischen* Aspekte der Schaltvoraussetzung werden hingegen mit Hilfe der *zweiten* Aktivierungsbedingung überprüft. Sie ist genau dann erfüllt, wenn der Ausdruck, der aus der Substitution der Transitionsannotation $AN_{TR}(tr_n)$ mit Hilfe von θ hervorgeht, in der statischen Struktur $A_{SIG_{ST}}$ gültig ist.

Die Schaltvoraussetzung für eine Integritätstransition tr_n unter einer Markierung M_r und einer Grundsubstitution θ kann durch konjunktive Verknüpfung der Aktivierungsbedingungen wie folgt formuliert werden:

$$\begin{array}{l}\forall tr_n \in TR_{INT},\ \theta \in 2^{\theta},\ M_r \in ZMF_{ON}: \\ AKT_{ON}\left(tr_n,\theta,M_r\right) :\Leftrightarrow \\ \qquad \forall st_m \in VB_{TR}(tr_n): \left(AN_{IR}\left(st_m,tr_n\right)[\theta]\right) \subseteq M_r(st_m) \wedge \\ \qquad A_{SIG_{ST}} \vDash \left(AN_{TR}(tr_n)[\theta]\right)\end{array}$$

Für eine Inferenztransition $tr_n \in TR_{INF}$ gilt unter einer Grundsubstitution $\theta \in 2^{\theta}$ und einer zulässigen Markierung $M_r \in ZMF_{ON}$ genau dann $AKT_{ON}(tr_n,\theta,M_r)$, wenn

(1.) $\forall st_m \in VB_{TR}(tr_n)$: $(AN_{IR}(st_m,tr_n)[\theta]) \subseteq M_r(st_m)$,

(2.) $A_{SIG_{ST}} \vDash (AN_{TR}(tr_n)[\theta])$,

(3.) $\forall st_m \in NB_{TR}(tr_n)$: $(AN_{FR}(tr_n,st_m)[\theta]) \not\subseteq M_r(st_m)$,

(4.) $\forall st_m \in NB_{TR}(tr_n)$: $\#M_r(st_m) \leq (KP(st_m) - \#(AN_{FR}(tr_n,st_m)))$ und

(5.) $\neg\exists tr_x \in TR_{INT},\theta_y \in 2^{\theta}$: $AKT_{ON}(tr_x,\theta_y,M_r)$

gelten.

Die erste und die zweite Aktivierungsbedingung für Inferenztransitionen stimmen mit den beiden Aktivierungsbedingungen für Integritätstransitionen überein. So wird mit der ersten Aktivierungsbedingung für Inferenztransitionen gewährleistet, dass die Marken, die entsprechend der Kantenannotation $AN_{IR}(st_m,tr_n)$ bei einer Grundsubstitution θ von der Inferenztransition eingelesen werden müssen, auf der Stelle st_m enthalten sind. Bezogen auf eine Inferenzregel $F \in INF_{SIG_{OS}}$ mit $trans_{INF}(INF)=tr_n$ bedeutet das, dass die Gültigkeit aller Formeln aus der Antezedenz-Komponente von F – repräsentiert durch die Markierungen der Stellen aus dem Vorbereich $VB_{TR}(tr_n)$ von tr_n – vorliegen muss, um auf die Gültigkeit der Subjugats-Komponente von F – repräsentiert durch die Markierungen der Stellen im Nachbereich $NB_{TR}(tr_n)$ von tr_n – schließen zu können.

Über die beiden Aktivierungsbedingungen, die bereits für Integritätstransitionen Gültigkeit haben, werden drei weitere Aktivierungsbedingungen für Inferenztransitionen geltend gemacht.

Mit der dritten Aktivierungsbedingung wird überprüft, ob die Marke, die entsprechend der Grundsubstitution θ auf der einzigen[1)] Nachbereichsstelle st_m abgelegt werden müsste, bereits auf dieser enthalten ist. Wenn die Marke bereits auf der Nachbereichsstelle st_m enthalten ist, ist das zu erschließende Wissen bereits in der Markierung des Ontolo-

1) Entsprechend den syntaktischen Anforderungen zur Formulierung von Inferenzregeln darf die Subjugatskomponente einer Inferenzregel F nur ein positives oder negatives Literal lit_{n+1} aufweisen; vgl. Abschnitt 3.1.4.1. Dieses Literal lit_{n+1} wird bei der Transformation von Inferenzregeln zu Inferenztransitionen derart berücksichtigt, dass eine Stelle st_m im Nachbereich $NB_{TR}(tr_n)$ von tr_n spezifiziert wird. Handelt es sich bei lit_{n+1} um ein positives Literal der Form $R_j(t_1,...,t_n)$, wird eine positive Stelle $st_m \in ST_{POS}$ spezifiziert. Handelt es sich hingegen bei lit_{n+1} um ein negatives Literal der Form $\neg R_j(t_1,...,t_n)$, wird eine negative Stelle $st_m \in ST_{NEG}$ spezifiziert.

gie-Netzes enthalten. Ansonsten ginge die Gültigkeit der Formel, die durch Inferenztransitionen erschließbar wäre, bereits aus der Markierung der Stelle st_m hervor. In diesem Fall kann eine Inferenztransition nicht aktiviert werden.

Die dritte Aktivierungsbedingung für Inferenztransitionen ist auf die *Inhalte* der Marken bezogen, die durch das Schalten einer Inferenztransition tr_n auf ihrer Ausgangsstelle $st_m \in NB_{TR}(tr_n)$ abgelegt würden. Komplementär hierzu wird mit der vierten Aktivierungsbedingung für Inferenztransitionen die *Anzahl* der Marken überprüft, die durch das Schalten von tr_n auf der Stelle[1)] st_m abgelegt würden. Wenn durch das Schalten einer Inferenztransition tr_n in einer Markierung M_r die Kapazität $KP(st_m)$ der Stelle $st_m \in NB_{TR}(tr_n)$ aufgrund der stellenbezogenen Markierung $M_r(st_m)$ und der Menge $\#(AN_{FR}(tr_n,st_m)[\theta])$ der auf st_m abzulegenden Marken überschritten würde, kann tr_n nicht aktiviert werden.

Mit der letzten Aktivierungsbedingung für Inferenztransitionen wird schließlich deren Unterordnung gegenüber der Aktivierung von Integritätstransitionen zum Ausdruck gebracht. In einem Ontologie-Netz ON kann demnach keine Inferenztransition tr_1 unter einer Markierung M_r aktiviert werden, solange mindestens eine Integritätstransition tr_2 unter einer beliebigen Grundsubstitution θ_y aktiviert ist.[2)] Dadurch wird gewährleistet, dass unter keinen Umständen aufgrund einer Markierung M_r, die bezüglich mindestens einer Integritätstransition tr_x unzulässig ist, mit Hilfe einer Inferenztransition tr_n auf weitere Fakten geschlossen werden kann, indem tr_n schaltet. Hierbei werden ausdrücklich nicht nur solche Integritätstransitionen berücksichtigt, deren Nachbarschaft $NA_{TR}(tr_x)$ sich mit der Nachbarschaft $NA_{TR}(tr_n)$ der Inferenztransition tr_n überschneidet.

Als bijektive Formel lässt sich die Aktivierung einer Inferenztransition tr_n unter einer Markierung M_r unter einer Grundsubstitution θ wie folgt ausdrücken:

1) Da in der vierten Aktivierungsbedingungen auf die allgemeine Flusskantenannotation AN_{FR} Bezug genommen wird, könnte sich die Quantifizierung auch über die Menge ST aller Stellen erstrecken. Die vierte Aktivierungsbedingung ist jedoch bewusst auf die Stellen in der Nachbarschaft $NB_{TR}(tr_n)$ eingeschränkt, da Stellen, die nicht in der unmittelbaren lokalen Umgebung von tr_n liegen, für die Aktivierung von Transitionen irrelevant sind.

2) Zu beachten ist, dass die Substitution θ_y, die für die Aktivierung der Integritätstransition tr_x vorausgesetzt wird, nicht mit der Substitution θ übereinstimmen muss, die für die Aktivierung der Inferenztransition tr_n vorausgesetzt wird.

$$
\begin{aligned}
&\forall tr_n \in TR_{INF},\ \theta \in 2^{\theta},\ M_r \in ZMF_{ON}:\\
&AKT_{ON}\left(tr_n,\theta,M_r\right) :\Leftrightarrow\\
&\qquad \forall st_m \in VB_{TR}(tr_n): \left(AN_{IR}\left(st_m,tr_n\right)[\theta]\right) \subseteq M_r(st_m) \wedge\\
&\qquad A_{SIG_{ST}} \models \left(AN_{TR}(tr_n)[\theta]\right) \wedge\\
&\qquad \forall st_m \in NB_{TR}(tr_n): \quad \left(AN_{FR}\left(tr_n,st_m\right)[\theta]\right) \not\subseteq M_r(st_m) \wedge\\
&\qquad \forall st_m \in NB_{TR}\left(tr_n\right): \quad \#M_r\left(st_m\right) \leq \left(KP\left(st_m\right) - \#\left(AN_{FR}\left(tr_n,st_m\right)\right)\right) \wedge\\
&\qquad \neg\exists tr_x \in TR_{INT}, \theta_y \in 2^{\theta}: AKT_{ON}\left(tr_x,\theta_y,M_r\right).
\end{aligned}
$$

Eine prozedurale Transition $tr_n \in TR_{PROZ}$ ist in einem Ontologie-Netz ON unter einer Grundsubstitution θ und einer Markierung M_r genau dann aktiviert, wenn

(1.) $\forall st_m \in VB_{TR}(tr_n)$: $(AN_{IR}(st_m,tr_n)[\theta]) \subseteq M_r(st_m)$,

(2.) $A_{SIG_{ST}} \models (AN_{TR}(tr_n)[\theta])$,

(3.) $\forall st_m \in VB_{TR}(tr_n)$: $(AN_{FR}(st_m,tr_n)[\theta]) \subseteq M_r(st_m)$,

(4.) $\forall\ st_m \in NA_{TR}(tr_n)$:
$\#(AN_{FR}(st_m,tr_n)) \leq \#M_r(st_m) \leq (KP(st_m) + \#(AN_{FR}(st_m,tr_n)) - \#(AN_{FR}(tr_n,st_m)))$ und

(5.) $\neg\exists tr_x \in TR_{DEK}$, $\theta_y \in 2^{\theta}$: $AKT_{ON}(tr_x,\theta_y,M_r)$

gelten. Die Schaltvoraussetzungen für prozedurale Transitionen umfassen die Aktivierungsbedingungen (1.) und (2.), die auch für deklarative Transitionen gelten. Darüber hinaus gelten drei weitere Aktivierungsbedingungen für prozedurale Transitionen.

Da prozedurale Transitionen – im Gegensatz zu deklarativen Transitionen – auch Marken von Stellen abziehen können, muss überprüft werden, ob die bei der Grundsubstitution θ zu entziehenden Marken auch auf den entsprechenden Stellen $st_m \in VB_{TR}(tr_n)$ enthalten sind. Hierfür ist die Aktivierungsbedingung (3.) vorgesehen. Sie ist genau dann erfüllt, wenn die Marken, die beim Schalten der prozeduralen Transition tr_n unter der Grundsubstitution θ durch eine inzidente Flusskante (st_m,tr_n) von der Stelle st_m abfließen müssten, auch auf st_m enthalten sind.

Mit der vierten Aktivierungsbedingung wird gewährleistet, dass nach dem Schalten einer Inferenztransition tr_n unter einer Referenzmarkierung M_r keine Folgemarkierung M_f eintreten kann, die bezüglich der Integritätsregel $\widehat{IB_{ON_4}}$ unzulässig ist. Eine Markierung M_f ist genau dann bezüglich der Integritätsbedingung $\widehat{IB_{ON_4}}$ unzulässig, wenn sie mindestens eine stellenbezogene Markierung $M_f(st_m)$ umfasst, für die $M_f < 0$ oder $M_f > K(st_m)$ gilt.

Die Aktivierungsbedingung (4.) kann in die drei Aktivierungsteilbedingungen

(4.1.) $\forall st_m \in VB_{TR}(tr_n)$:
$\#M_r(st_m) \geq \#(AN_{FR}(st_m,tr_n))$,

(4.2.) $\forall st_m \in NB_{TR}(tr_n) \backslash VB_{TR}(tr_n)$:
$\#M_r(st_m) \leq (KP(st_m) - \#(AN_{FR}(tr_n,st_m)))$ und

(4.3.) $\forall st_m \in NB_{TR}(tr_n)) \cap VB_{TR}(tr_n)$:
$\#M_r(st_m) \leq (KP(st_m) + \#(AN_{FR}(st_m,tr_n)) - \#(AN_{FR}(tr_n,st_m)))$

unterteilt werden. Mit der ersten Teilbedingung (4.1.) wird gewährleistet, dass keine prozedurale Transition tr_n aktiviert werden kann, die von einer Stelle st_m mehr Marken abziehen würde als die Kardinalität $\#(M_r(st_m))$ der stellenbezogenen Markierung $M_r(st_m)$. Sie stellt insofern eine partiell redundante Teilaktivierungsbedingung dar, als dass die Erfülltheit der Aktivierungsbedingung (3.) auch stets die Erfülltheit der ersten Aktivierungsteilbedingung (4.1.) zufolge hat.

Mit der zweiten Teilbedingung (4.2.) wird gewährleistet, dass keine prozedurale Transition tr_n aktiviert werden kann, die auf einer Stelle st_m, die zwar im Nachbereich $NB_{TR}(tr_n)$, aber nicht im Vorbereich $VB_{TR}(tr_n)$ von tr_n vorkommt, auf st_m mehr Marken ablegen würde, als die stellenbezogene Kapazität $KP(st_m)$ erlaubt.

Mit der dritten Teilbedingung (4.3.) wird schließlich gewährleistet, dass die Netto-Markenerzeugung auf einer Stelle st_m, die sowohl im Vor- als auch im Nachbereich $VB_{TR}(tr_n)$ bzw. $NB_{TR}(tr_n)$ von tr_n vorkommt, nicht zu einer Überschreitung der stellenbezogenen Kapazität $K(st_m)$ führen würde.

Die drei Aktivierungsteilbedingungen (4.1.), (4.2.) und (4.3.) können zur Aktivierungsbedingung (4.) zusammengefasst werden, da für alle Knotenpaare (kn_x,kn_y), die nicht in der Flussrelation FR enthalten sind, $AN_{FR}(kn_x,kn_y)=\varnothing_M$ gilt.[1)]

Darüber hinaus unterscheiden sich die Schaltvoraussetzungen für prozedurale Transitionen von den Schaltvoraussetzungen für Inferenztransitionen hinsichtlich zweier Aktivierungsbedingungen. Erstens entfällt die vierte Aktivierungsbedingung für Inferenztransitionen, mit der festgelegt wurde, dass keine der Stellen aus dem Nachbereich $NB_{TR}(tr_n)$ der betrachteten Transition tr_n bereits die Marke aufweisen darf, die durch das Schalten der Transition auf diesen Stellen abgelegt würde. Im Gegensatz zu Inferenztransitionen können prozedurale Transitionen auch dann schalten, wenn auf den Stellen im Nachbereich bereits identische Kopien der abzulegenden Marke enthalten sind.

Zweitens wird mit der fünften Aktivierungsbedingung für prozedurale Transitionen eingefordert, dass im Zustand r keine deklarative Transition tr_x und keine Substitution θ_y existieren dürfen, so dass tr_x unter M_r und θ_y aktiviert ist. Durch diese Anforderung wird gewährleistet, dass keine prozedurale Transition tr_n schalten kann, solange eine deklarative Transition unter einer beliebigen Substitution θ_y aktiviert ist. Durch diese fünfte Schaltvoraussetzung wird die Aktivierbarkeit von prozeduralen Transitionen der Aktivierbarkeit aller deklarativen Transitionen untergeordnet.

1) Dies wurde im Rahmen der Definition der allgemeinen Kantenannotationsfunktion AN_{FR} unter Rückgriff auf die spezielle Kantenannotationsfunktion AN^*_{FR} festgelegt; vgl. Abschnitt 4.2.2.1.1.

Die Schaltvoraussetzungen für prozedurale Transitionen können durch die Konjunktion der Aktivierungsbedingungen wie folgt als bijektive Formel ausgedrückt werden:

$$
\begin{aligned}
&\forall tr_n \in TR_{PROZ},\ \theta \in 2^{\theta},\ M_r \in ZMF_{ON}:\\
&AKT_{ON}(tr_n,\theta,M_r) :\Leftrightarrow\\
&\qquad \forall st_m \in VB_{TR}(tr_n):\ \big((AN_{IR}(st_m,tr_n)[\theta]) \oplus (AN_{FR}(st_m,tr_n)[\theta])\big) \subseteq M_r(st_m) \wedge\\
&\qquad A_{SIG_{ST}} \models (AN_{TR}(tr_n)[\theta]) \wedge\\
&\qquad \forall st_m \in NA_{TR}(tr_n):\\
&\qquad\qquad \#(AN_{FR}(st_m,tr_n)) \leq \#M_r(st_m) \leq \big(KP(st_m) + \#(AN_{FR}(st_m,tr_n)) - \#(AN_{FR}(tr_n,st_m))\big) \wedge\\
&\qquad \neg\exists tr_x \in TR_{DEK}, \theta_y \in 2^{\theta}:\ AKT_{ON}(tr_x,\theta_y,M_r)
\end{aligned}
$$

Im Gegensatz zur Schaltvoraussetzung ist die Schalt*wirkung* für deklarative[1)] und prozedurale Transitionen identisch definiert. Wenn eine Transition $tr_n \in TR$ unter einer Substitution θ und einer Markierung M_r schaltet, werden

(1.) von allen Eingangsstellen $st_m \in VB_{TR}(tr_n)$ von tr_n entsprechend der Substitution $(AN_{FR}(st_m,tr_n)[\theta])$ Marken abgezogen und

(2.) auf allen Ausgangsstellen $st_m \in NB_{TR}(tr_n)$ von tr_n entsprechend der Substitution $(AN_{FR}(tr_n,st_m)[\theta])$ Marken abgelegt.

Formal wird die Schaltwirkung durch die Schaltfunktion

$$SR_{ON}: ZMF_{ON} \times 2^{\theta} \times TR \rightarrow ZMF_{ON}$$

bestimmt. Die Schaltfunktion SR_{ON} ordnet jedem Drei-Tupel (M_r,θ,tr_n), bestehend aus einer zulässigen Referenzmarkierung $M_r \in ZMF_{ON}$, einer Grundsubstitution $\theta \in 2^{\theta}$ und einer Transition $tr_n \in TR$, eine zulässige Folgemarkierung $SR_{ON}(M_r,\theta,tr_n)=M_f$ zu. Dabei sind die Bilder der stellenbezogenen Folgemarkierung $M_f(st_m)$ nach dem Schalten einer Transition tr_n unter einer Substitution θ und einer Referenzmarkierung M_r wie folgt bestimmt:

$$
M_f(st_m)=\begin{cases}
M_r(st_m)\setminus(AN_{FR}(st_m,tr_n)[\theta]) & \forall\ st_m \in \big(VB_{TR}(tr_n)\setminus NB_{TR}(tr_n)\big)\\
M_r(st_m)\oplus(AN_{FR}(tr_n,st_m)[\theta]) & \forall\ st_m \in \big(NB_{TR}(tr_n)\setminus VB_{TR}(tr_n)\big)\\
M_r(st_m)\setminus(AN_{FR}(st_m,tr_n)[\theta])\oplus(AN_{FR}(tr_n,st_m)[\theta]) & \forall\ st_m \in \big(NB_{TR}(tr_n)\cap VB_{TR}(tr_n)\big)\\
M_r(st_m) & \text{sonst}
\end{cases}
$$

Der Schaltakt einer Transition $tr_n \in TR$ hat lediglich Relevanz für die Markierungen der Stellen, die in der Nachbarschaft $NA_{TR}(tr_n)$ von tr_n liegen. Durch den (substitutionsspezifischen) Schaltakt einer Transition tr_n wird nämlich der Übergang von einer Referenzmarkierung M_r zu einer Folgemarkierung M_f bewirkt, von dem lediglich die Stellen

1) Wie bereits zuvor erörtert, sind von den deklarativen Transitionen in einem Ontologie-Netz lediglich die Schaltwirkungen von Inferenztransitionen von Interesse. Die Schaltwirkung von Integritätstransitionen interessiert nicht, da deren Aktiviertheit jeweils auf eine unzulässige Markierung hinweist.

im Vor- und Nachbereich von tr_n betroffen sind. Dadurch wird das *Lokalitätsprinzip*, das bereits für S/T-Netze angesprochen wurde,[1] auch für Ontologie-Netze beibehalten.

Wenn eine Transition tr_n unter einer Referenzmarkierung M_r und einer Grundsubstitution θ schaltet, wird zu der Folgemarkierung $SR_{ON}(M_r)=M_f$ übergegangen. Der Zusammenhang wird auch in der Form

$$M_r\ [(tr_n,\theta)>\ M_f$$

angegeben.

Die Multimenge ontologischer Termtupel, die durch das Schalten der Transition tr_n unter einer Grundsubstitution θ von einer Stelle $st \in VB_{TR}(tr_n)$ abgezogen wird, kann durch eine transitionsspezifische Löschfunktion

$$tr_n^-: 2^\theta \times ST \rightarrow MULT(GTT_{SIG_{OS}})$$

mit

$$tr_n^-(\theta,st_m)=\begin{cases} AN_{FR}(st_m,tr_n)[\theta]; & \text{wenn } st_m \in VB_{TR}(tr_n) \\ \varnothing_M; & \text{sonst} \end{cases}$$

angegeben werden. Wenn eine Transition tr_n unter einer Grundsubstitution θ und einer Markierung M_r schaltet, werden demnach brutto von einer Stelle $st \in VB_{TR}(tr_n)$ solche Kopien von Marken abgezogen, die im substituierten Ausdruck $AN_{FR}(st_m,tr_n)[\theta]$ enthalten sind.

Die Marken, die durch das Schalten einer Transition tr_n unter einer Grundsubstitution θ und einer Markierung M_r auf einer Stelle $st_m \in NB_{TR}(tr_n)$ abgelegt werden, werden mit Hilfe der transitionsspezifischen Erzeugungsfunktion

$$tr_n^+: 2^\theta \times ST \rightarrow MULT(GTT_{SIG_{OS}})$$

mit

$$tr_n^+(\theta,st_m)=\begin{cases} AN_{FR}(tr_n,st_m)[\theta]; & \text{wenn } st_m \in NB_{TR}(tr_n) \\ \varnothing_M; & \text{sonst} \end{cases}$$

bestimmt. Die transitionsspezifische Erzeugungsfunktion tr_n^+ ordnet jedem Tupel (θ,st_m) – bestehend aus einer Grundsubstitution $\theta \in 2^\theta$ und einer Stelle $st_m \in ST$ – eine Multimenge mult über der Menge $GTT_{SIG_{OS}}$ aller ontologischen Grundtermtupel zu. Wenn die betrachtete Stelle st_m im Nachbereich $NB_{TR}(tr_n)$ von tr_n liegt, beträgt die Brutto-Markenerzeugung durch tr_n auf st_m genau die Marken, die der mittels θ substituierten Kantenannotation $AN_{FR}(tr_n,st_m)$ zur adjazenten Kante (tr_n,st_m) entsprechen.

1) Vgl. Abschnitt 3.2.2.2.2.1

4.2.2.2.1.2 Schaltregel für Transitionsfolgen in Ontologie-Netzen

Die Schaltvoraussetzungen und -wirkungen aus Ontologie-Netzen wurden bislang lediglich für einzelne Transitionen untersucht. Analog zu der Vorgehensweise bei S/T-Netzen werden im Folgenden Schaltvoraussetzungen und -wirkungen von Transitions*folgen* thematisiert. Dadurch können auch solche Übergänge zwischen zustandsbezogenen Markierungen aufgezeigt werden, die nicht in einem unmittelbaren Ursache-Wirkungs-Zusammenhang stehen. Im Unterschied zu S/T-Netzen müssen allerdings bei Ontologie-Netzen auch stets die Grundsubstitutionen berücksichtigt werden, unter denen die Transitionen aus der betrachteten Transitionsfolge schalten.[1)] Somit ergeben sich im Vergleich zu Schaltfolgen aus S/T-Netzen anspruchsvollere Konstruktionen.

Die Menge $(TR\times 2^{\theta})^*$ umfasst alle Folgen von Zwei-Tupeln, die jeweils aus einer Transition $tr_n \in TR$ und einer Grundsubstitution $\theta \in 2^{\theta}$ bestehen. Die Elemente der Menge $(TR\times 2^{\theta})^*$ sind induktiv wie folgt definiert:

(1.) $\lambda \in (TR\times 2^{\theta})^*$,

(2.) $\forall\ tr_n \in TR,\ \theta \in 2^{\theta}:\ (tr_n,\theta) \in (TR\times 2^{\theta})^*$ und

(3.) $\forall\ sf \in (TR\times 2^{\theta})^*,\ tr_n \in TR,\ \theta \in 2^{\theta} :\ (sf\ (tr_n,\theta)) \in (TR\times 2^{\theta})^*$.

Jedes Element $sf \in (TR\times 2^{\theta})^*$ wird als *höhere Schaltfolge* bezeichnet und in der Form $sf=(tr_1,\theta_1)...(tr_{len_{ON}(sf)},\theta_{len_{ON}(sf)})$ mit $tr_x \in TR$ und $\theta_x \in 2^{\theta}$ für $x=1,...,len_{ON}(sf)$ und $len_{ON}(sf) \in (\mathbb{N}_+ \cup \{\infty\})$ angegeben.[2)] Das Element λ entspricht der Nullschaltfolge, in der keine Transition schaltet. Zudem kann in einer höheren Schaltfolge $sf \in (TR\times 2^{\theta})^*$ eine Transition tr_n mehrfach – auch in Kombination mit unterschiedlichen Grundsubstitutionen – vorkommen.

Die Menge $(TR\times 2^{\theta})^*$ aller höheren Schaltfolgen kann hinsichtlich der *Länge* der Elemente in die zueinander disjunkten Teilmengen

$$(TR\times 2^{\theta})^* = (TR\times 2^{\theta})^0 \cup (TR\times 2^{\theta})^1 \cup ... \cup (TR\times 2^{\theta})^{\infty}$$

unterteilt werden. Jede Menge $(TR\times 2^{\theta})^n$ mit $n \in \mathbb{N}$ umfasst höhere Schaltfolgen der Länge n und wird wie folgt definiert:

(1.) $(TR\times 2^{\theta})^0 = \lambda$ und

(2.) $(TR\times 2^{\theta})^n = \{sf\ (tr_n,\theta)\ |\ sf \in TR^{n-1} \wedge tr_n \in TR \wedge \theta \in 2^{\theta} \text{ für } n \in \mathbb{N}_+\}$.

Die Länge n einer höheren Schaltfolge $sf \in (TR\times 2^{\theta})^*$ wird durch die Funktion

$$len_{ON}: (TR\times 2^{\theta})^* \rightarrow (\mathbb{N} \cup \{\infty\})$$

---

1) Zur Berücksichtigung von Substitutionen in Schaltfolgen höherer Petri-Netze vgl. ERWIN (2002), S. 154.

2) Die Funktion len_{ON} wird – analog zu der Funktion len_{ST} – dazu verwendet, die Länge einer höheren Schaltfolge zu bestimmen. Sie wird weiter unten näher bestimmt.

bestimmt, wobei

(1.) $len_{ON}(\lambda)=0$,

(2.) $len_{ON}(tr_n,\theta)=1$ für $tr_n \in TR$ und $\theta \in 2^{\theta}$ und

(3.) $len_{ON}(sf\,(tr_n,\theta))=len_{ON}(sf)+1$, wenn $sf \in (TR \times 2^{\theta})^*$, $tr_n \in TR$ und $\theta \in 2^{\theta}$

gelten. Für die längenspezifischen Mengen höherer Schaltfolgen gilt somit:

$$TR^n=\{sf \mid sf \in (TR \times 2^{\theta})^* \wedge len_{ON}(sf)=n\} \text{ für } n \in \mathbb{N}.$$

Eine höhere Schaltfolge $sf \in (TR \times 2^{\theta})^*$ mit $sf=(tr_1,\theta_1)\ldots(tr_{len_{ON}(sf)},\theta_{len_{ON}(sf)})$ ist genau dann unter eine Markierung M_r aktiviert, wenn eine derartige Folge $M_r \ldots M_f$ von Markierungen existiert, so dass

$$AKT_{ON}(tr_1,\theta_1,M_r) \wedge M_r\,[(tr_1,\theta_1)\!>M_{z_1} \wedge \ldots \wedge$$
$$AKT_{ON}(tr_{len_{ON}(sf)},\theta_{len_{ON}(sf)},M_{z_{lenON(sf)-1}}) \wedge M_{z_{lenON(sf)-1}}\,[tr_{len_{ON}(sf)},\theta_{len_{ON}(sf)}\!>M_f$$

gilt. Um die Aktivierung einer höheren Schaltfolge $sf \in (TR \times 2^{\theta})^*$ mit $sf=(tr_1,\theta_1)\ldots(tr_{len_{ON}(sf)},\theta_{len_{ON}(sf)})$ zu repräsentieren, wird der Ausdruck $AKT_{ON}^{SF}(sf)$ mit

$$\boxed{\begin{gathered}
\forall\Big((tr_1,\theta_1)\ldots\big(tr_{len_{ON}(sf)},\theta_{len_{ON}(sf)}\big)\Big)\in\big(TR\times 2^{\theta}\big)^*,\ M_r,M_f\in ZMF_{ON}:\\
AKT_{ON}^{SF}\Big((tr_1,\theta_1)\ldots\big(tr_{len_{ON}(sf)},\theta_{len_{ON}(sf)}\big),M_r\Big)\ :\Leftrightarrow\\
\exists M_{z_1},\ldots,M_{z_{len_{ON}(sf)-1}}\in ZMF_{ON}\\
\Big(AKT_{ON}\big((tr_1,\theta_1),M_r\big)\ \wedge\ SR_{ON}\big((tr_1,\theta_1),M_r\big)=M_{z_1}\Big)\wedge\ldots\wedge\\
\Big(AKT_{ON}\big(\big(tr_{len_{ON}(sf)},\theta_{len_{ON}(sf)}\big),M_{z_{len_{ON}(sf)-1}}\big)\ \wedge\ SR_{ON}\big(\big(tr_{len_{ON}(sf)},\theta_{len_{ON}(sf)}\big),M_{z_{len_{ON}(sf)-1}}\big)=M_f\Big)
\end{gathered}}$$

verwendet.

Wenn eine höhere Schaltfolge $(tr_1,\theta_1)\ldots(tr_{len_{ON}(sf)},\theta_{len_{ON}(sf)})$ unter einer Markierung M_r aktiviert ist, können die Transitionen $tr_1,\ldots,tr_{len_{ON}(sf)}$ nacheinander unter der jeweiligen Grundsubstitution $\theta_1,\ldots,\theta_{len_{ON}(sf)}$ schalten. So kann die erste Transition tr_1 unter der Grundsubstitution θ_1 schalten. Wenn tr_1 in der *Startmarkierung* M_r unter der Grundsubstitution θ_1 schaltet, wird zu der *Zwischenmarkierung* M_{z_1} übergegangen. In M_{z_1} kann die Transition tr_2 unter der Grundsubstitution θ_2 schalten.[1)] In der letzten Zwischenmarkierung $M_{z_{lenON(sf)-1}}$ kann die letzte Transition $tr_{len_{ON}(sf)}$ unter der Grundsubstitution $\theta_{len_{ON}(sf)}$ schalten. Wenn $tr_{len_{ON}(sf)}$ unter $M_{z_{lenON(sf)-1}}$ und $\theta_{len_{ON}(sf)}$ schaltet, wird von $M_{z_{lenON(sf)-1}}$ zu der *Schlussmarkierung* M_f übergegangen.

Die Schaltfunktion SF_{ON} zur Bestimmung der Folgemarkierung M_f, die durch das Schalten einer Transition tr_n unter einer Grundsubstitution θ aus einer Referenzmarkie-

---

1) Zu beachten ist, dass sich weder die Transition tr_2 von der Transition tr_1 noch die Substitution θ_2 von der Substitution θ_1 unterscheiden müssen.

rung M_r hervorgeht, kann kanonisch auf höhere Schaltfolgen übertragen werden. Die partielle Schaltfunktion

$$SR_{ON}^{SF}: \left(TR \times 2^{\theta}\right)^* \times ZMF_{ON} \rightarrow ZMF_{ON}$$

bildet ein Tupel (sf,M_r) – bestehend aus einer höheren Schaltfolge $sf \in (TR \times 2^{\theta})^*$ mit $sf=(tr_1,\theta_1)...(tr_{len_{ON}(sf)},\theta_{len_{ON}(sf)})$ und einer zulässigen Startmarkierung $M_r \in ZMF_{ON}$ – auf eine zulässige Schlussmarkierung $M_f \in ZMF_{ON}$ ab, wenn $AKT_{ON}^{SF}\left(sf,M_r\right)$ gilt. Die Schlussmarkierung M_f wird hierbei wie folgt bestimmt:

$$\begin{gathered}
\forall \left(\left(tr_1,\theta_1\right)...\left(tr_{len_{ON}(sf)},\theta_{len_{ON}(sf)}\right)\right) \in \left(TR \times 2^{\theta}\right)^*;\ M_r, M_{z_1}, ..., M_{z_{len_{ON}(sf)-1}}, M_f \in ZMF_{ON}: \\
SR_{ON}^{SF}\left(\left(tr_1,\theta_1\right)...\left(tr_{len_{ON}(sf)},\theta_{len_{ON}(sf)}\right), M_r\right) = M_f \ :\Leftrightarrow \\
SR_{ON}\left(\left(tr_1,\theta_1\right),M_r\right) = M_{z_1} \ \wedge ... \wedge \ SR_{ON}\left(\left(tr_{len_{ON}(sf)},\theta_{len_{ON}(sf)}\right), M_{z_{len_{ON}(sf)-1}}\right) = M_f
\end{gathered}$$

Eine Folge

$$\begin{gathered}
\left(M_r,\left(tr_1,\theta_1\right),M_{z_1}\right)\left(M_{z_1},\left(tr_2,\theta_2\right),M_{z_2}\right)... \\
\left(M_{z_{len_{ON}(sf)-2}},\left(tr_{len_{ON}(sf)-1},\theta_{len_{ON}(sf)-1}\right),M_{z_{len_{ON}(sf)-1}}\right)\left(M_{z_{len_{ON}(sf)-1}},\left(tr_{len_{ON}(sf)},\theta_{len_{ON}(sf)}\right),M_f\right) \\
\in \left(ZMF_{ON} \times (TR \times 2^{\theta}) \times ZMF_{ON}\right)^* \\
\\
\text{mit } M_r \ [\left(tr_1,\theta_1\right)> M_{z_1} \\
\forall x \in \{2,...,len_{ON}(sf)\}: M_{z_{x-1}} \ [\left(tr_x,\theta_x\right)> M_{z_x} \\
\text{und } M_{z_{len_{ON}(sf)-1}} \ [\left(tr_{len_{ON}(sf)},\theta_{len_{ON}(sf)}\right)> M_f
\end{gathered}$$

wird als *höherer formaler Prozess* in bezeichnet. Ein höherer formaler Prozess in einem Ontologie-Netz ist eine Folge von Tupeln in der Form *Start- oder Zwischenmarkierung* – (*Transition*, *Grundsubstitution*) – *Zwischen- oder Schlussmarkierung*. Dabei ist jede Folgemarkierung aus einem Tupel jeweils die Referenzmarkierung für das folgende Tupel.

Wenn die Schaltfolge $sf \in (TR \times 2^{\theta})^*$ unter der Startmarkierung M_r aktiviert ist und die Schlussmarkierung M_f durch das Schalten der Schaltfolge $sf \in (TR \times 2^{\theta})^*$ mit $sf=(tr_1,\theta_1)...(tr_{len_{ON}(sf)},\theta_{len_{ON}(sf)})$ aus der Startmarkierung M_r hervorgeht, wird hierfür die *schaltfolgenspezifische Erreichbarkeitsrelation*

$$[sf> \subseteq ZMF_{ON} \times ZMF_{ON}$$

verwendet.[1)] Eine Schlussmarkierung M_f geht durch das Schalten einer höheren Schaltfolge sf aus einer Startmarkierung M_r genau dann hervor, wenn

(1.) $sf=\lambda \wedge M_r=M_f$ oder

(2.)

$$
\begin{aligned}
\exists M_{z_{len_{ON}(sf)-1}} \in ZMF_{ON}:\quad & AKT_{ON}^{SF}\left(\left(tr_1,\theta_1\right)\ldots,\left(tr_{len_{ON}(sf)-1},\theta_{len_{ON}(sf)-1}\right)M_r\right)\wedge\\
& M_r[\left(tr_1,\theta_1\right)\ldots,\left(tr_{len_{ON}(sf)-1},\theta_{len_{ON}(sf)-1}\right)>M_{z_{len_{ON}(sf)-1}}\wedge\\
& AKT_{ON}\left(\left(tr_{len_{ON}(sf)},\theta_{len_{ON}(sf)}\right),M_{z_{len_{ON}(sf)-1}}\right)\wedge\\
& M_{z_{len_{ON}(sf)-1}}[\left(tr_{len_{ON}(sf)},\theta_{len_{ON}(sf)}\right)>M_f
\end{aligned}
$$

gelten.

---

1) Zu beachten ist, dass sich die schaltfolgenspezifische Erreichbarkeitsrelation [sf> in Ontologie-Netzen von der entsprechenden Relation in S/T-Netzen dadurch unterscheidet, dass sich [sf> auf höhere Schaltfolgen bezieht. In höheren Schaltfolgen wird wiederum – im Gegensatz zu den Schaltfolgen aus S/T-Netzen – zu jeder Transition tr_n, die darin vorkommt, die entsprechende Substitution θ_x aufgeführt wird, unter der tr_n schaltet.

4.2.2.2.2 Nebenläufigkeit und Konflikte

Die Bestimmung der entweder[1)] nebenläufigen oder konfliktionären Aktiviertheit von Transitionen aus S/T-Netzen kann analog auf Transitionen in Ontologie-Netzen übertragen werden. Da die Aktiviertheit von Transitionen aus Ontologie-Netzen immer in Bezug auf eine Grundsubstitution θ definiert ist, müssen allerdings bei der Bestimmung, ob eine Nebenläufigkeit oder ein Konflikt vorliegt, auch die Grundsubstitutionen angegeben werden, unter denen Transitionen aktiviert sind. Somit kann es in Ontologie-Netzen durchaus sein, dass zwei Transitionen tr_1 und tr_2 bezüglich zweier Grundsubstitution θ_{1_1} bzw. θ_{2_1} nebenläufig und bezüglich zweier weiterer Grundsubstitutionen θ_{1_2} bzw. θ_{2_2} konfliktionär aktiviert sind.

Eine konfliktionäre Aktivierung von zwei Transitionen tr_1 und tr_2 unter den Grundsubstitutionen θ_1 bzw. θ_2 liegt vor, wenn das Schalten einer der beiden Transitionen unter der entsprechenden Grundsubstitution die Aktivierung der jeweils anderen Transition unter der entsprechenden Grundsubstitution aufheben würde. Die Aktivierung einer Transition tr_2 kann durch das Schalten einer Transition tr_1 nur dann aufgehoben werden, wenn $NA_{TR}(tr_1) \cap NA_{TR}(tr_2) \neq \varnothing$ gilt. Insofern ist es für das Vorliegen eines Konflikts zwischen zwei Transitionen *notwendig*, dass mindestens eine Stelle st_m sowohl in der Nachbarschaft $NA_{TR}(tr_1)$ der Transition tr_1 als auch in der Nachbarschaft $NA_{TR}(tr_2)$ der Transition tr_2 vorkommt. Hinreichend für das Vorliegen eines Konflikts ist die Aufhebung der Aktivierung der jeweils anderen Transition durch das Schalten einer der Transitionen.[2)] Gelten hingegen $NA_{TR}(tr_1) \cap NA_{TR}(tr_2) = \varnothing$, $AKT_{ON}(tr_1,M_r)$ und $AKT_{ON}(tr_2,M_r)$, so sind die hinreichenden Bedingungen für die nebenläufige Aktivierung beider Transitionen tr_1 und tr_2 erfüllt.

Bei der Untersuchung von Nebenläufigkeit und Konflikt in Ontologie-Netzen werden nur solche Integritätsverletzungen untersucht, die bezüglich der metasprachlichen Integritätsbedingung $\widehat{IB_{ON_4}}$ eintreten würden, wenn die beiden Transitionen tr_1 und tr_2 gemeinsam schalten würden. Objektsprachliche Integritätsverletzungen, die aufgrund der Aktivierbarkeit einer Integritätstransition in der Folgemarkierung M_f vorliegen können, werden hingegen nicht betrachtet. Ein Beispiel für den objektsprachlichen Konflikt zwischen Transitionen ist in der folgenden Abbildung illustriert.

1) Nebenläufigkeit und Konflikt stellen auch im Fall von Ontologien-Netze gegenseitig ausschließende und den Raum aller Aktivierungsmöglichkeiten ausschöpfende Möglichkeiten der Aktivierung von Transitionen dar. Daher impliziert die Aktivierung von Transitionen unter der einen Möglichkeit den Ausschluss der jeweils anderen Möglichkeit. Je nach Argumentationskontext wird daher im Folgenden die für die Argumentation nahe liegende Aktivierungsmöglichkeit thematisiert.

2) Die Aufhebung der Aktivierung einer Transition tr_2 durch das Schalten einer Transition tr_1 ist zwar hinreichend, jedoch nicht notwendig für das Vorliegen eines Konflikts zwischen tr_1 und tr_2. Beispielsweise kann es im Fall von Schleifen sein, dass ein Konflikt zwischen zwei Transitionen tr_1 und tr_2 vorliegt, obwohl das Schalten von einer der Transitionen die Aktivierung der jeweils anderen nicht aufhebt. Die Besonderheiten, die für Transitionen aus Schleifen gelten, werden weiter unten thematisiert.

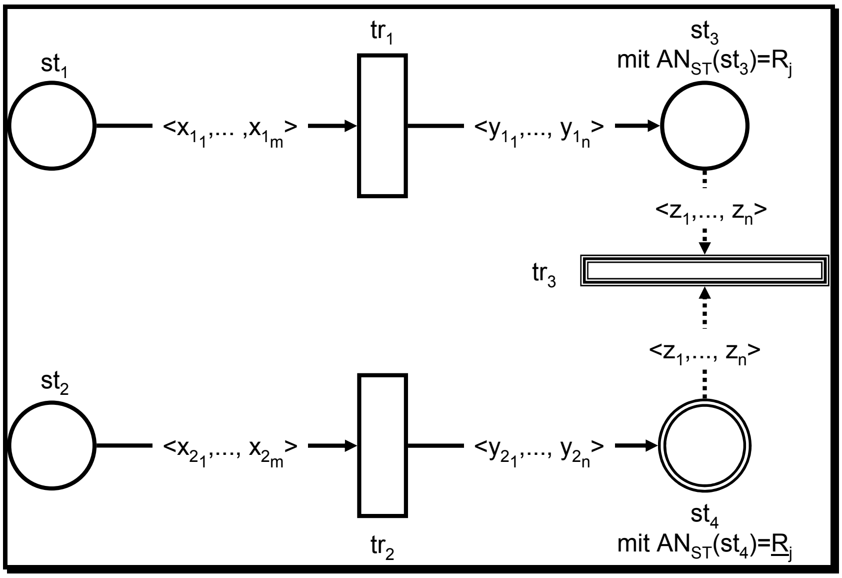

Abbildung 29: Objektsprachlicher Konflikt zwischen Transitionen

Wenn z.B. durch das Schalten der Transitionen tr_1 und tr_2 unter den Grundsubstitutionen θ_1 bzw. θ_2 die gleiche Marke ($<y_{1_1},...,y_{1_n}>[\theta_1]=<y_{2_1},...,y_{2_n}>[\theta_2]$) auf den Stellen st_3 und st_4 abgelegt würde, dann sind tr_1 und tr_2 objektsprachlich konfliktionär aktiviert, da unter der Folgemarkierung M_f die Integritätstransition tr_3 aktiviert wäre. Sie wird aktiviert, da die Stellen st_3 und st_4 mit den Relationssymbolen R_j bzw. $\underline{R}_j$ annotiert sind, welche stets überlappungsfreie Extensionen aufweisen müssen. Eine solche Integritätsverletzung würde durch die Aktivierung der Integritätstransition tr_3 aufgedeckt werden.

Die nebenläufige oder konfliktionäre Aktivierung von zwei Transitionen in Ontologie-Netzen setzt stets deren Aktivierung voraus. Aufgrund der zuvor aufgestellten Aktivierungsprioritäten, denen zufolge die Aktivierung prozeduraler Transitionen solange blockiert wird, wie deklarative Transitionen aktiviert werden können, kann eine nebenläufige oder konfliktionäre Aktivierung nur entweder zwischen deklarativen oder nur zwischen prozeduralen Transitionen bestehen. Die nebenläufige oder konfliktionäre Aktivierung von einer deklarativen und einer prozeduralen Transition ist in Ontologie-Netzen grundsätzlich nicht möglich.

Ein Konflikt zwischen zwei deklarativen Transitionen kann nur im Fall von Inferenztransitionen vorliegen, da nur für solche die Schaltwirkung relevant ist. Für Integritätstransitionen werden Schaltakte nicht thematisiert. Schaltakte sind lediglich für Inferenztransitionen erlaubt. In diesem Fall kann ein Konflikt zwischen zwei Inferenztransitionen tr_1 und tr_2 vorliegen, wenn mindestens eine Stelle st_m sowohl in $NB_{TR}(tr_1)$ als auch in $NB_{TR}(tr_2)$ vorkommt. Gemeinsame Stellen in den Vorbereichen von Transitionen haben keinen Einfluss auf die Frage der Nebenläufigkeit, da Inferenztransitionen stets marken*erzeugender*, aber niemals marken*verbrauchender* Natur sind. Mit den Stellen in ihren Vorbereichen sind Inferenztransitionen grundsätzlich mit Informationskanten verbunden.

Ein Ontologie-Teilnetz mit potenziellem Konflikt zwischen Inferenztransitionen ist in der folgenden Abbildung dargestellt:

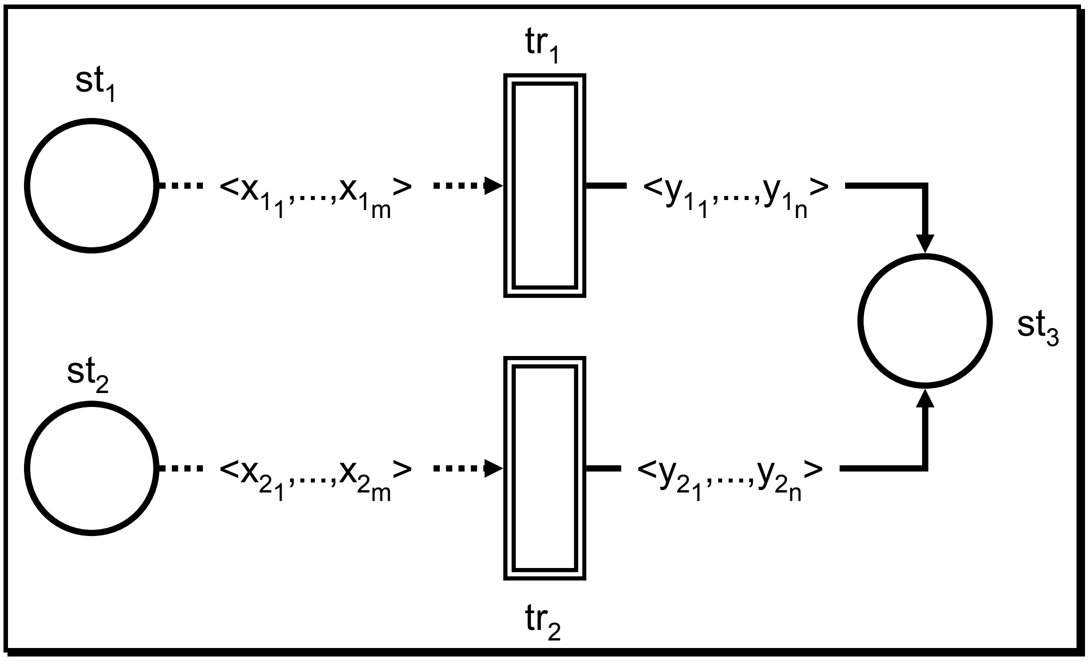

Abbildung 30: Potenzieller Konflikt zwischen Inferenztransitionen

Angenommen sei dabei, dass $AKT_{ON}(tr_1,\theta_1,M_r)$ und $AKT_{ON}(tr_2,\theta_2,M_r)$ gelten. Die beiden Aktivierungen sind genau dann konfliktionär, wenn tr_1 und tr_2 derart aktiviert sind, dass

(1.) entweder voneinander unterschiedliche Marken
$(<y_{1_1},\ldots,y_{1_n}>[\theta_1]\neq<y_{2_1},\ldots,y_{2_n}>[\theta_2])$
auf st_3 abgelegt werden müssten, allerdings dadurch die stellenbezogene Folgemarkierung
$M_f(st_3)=M_r(st_3) \oplus <y_{1_1},\ldots,y_{1_n}>[\theta_1]+<y_{2_1},\ldots,y_{2_n}>[\theta_2]$
eine größere Kardinalität $\#(M_f(st_3))$ aufweisen würde als durch die stellenbezogene Kapazität $KP(st_3)$ erlaubt ist,

(2.) oder beide Transitionen die gleiche Marke
$(<y_{1_1},\ldots,y_{1_n}>[\theta_1]=<y_{2_1},\ldots,y_{2_n}>[\theta_2])$
auf st_3 ablegen würden.

Im ersten Fall liegt ein Konflikt vor, da das Schalten einer der beiden Transitionen dazu führen würde, dass die Kapazität $KP(st_3)$ der Stelle ausgeschöpft wird, so dass die andere Transition nicht aktiviert werden kann. Obwohl durch die nicht aktivierbare Transition Wissen erschlossen werden könnte, das implizit in der Referenzmarkierung M_r enthalten ist, kann sie nicht schalten. Im zweiten Fall kann die andere Transition nicht aktiviert werden, weil das Wissen, das durch sie erschließbar wäre, bereits durch die geschaltete Transition erschlossen wurde.

Zwei prozedurale Transitionen tr_1 und tr_2 sind in einer Referenzmarkierung M_r unter den Grundsubstitutionen θ_1 bzw. θ_2 genau[1)] dann nebenläufig aktiviert, wenn

(1.) $AKT_{ON}(tr_1,\theta_1,M_r)$ und $AKT_{ON}(tr_2,\theta_2,M_r)$ gelten und

(2.) jede stellenbezogene Markierung $M_r(st_m)$ alle Marken umfasst, die durch den gemeinsamen Markenkonsum $AN_{FR}(st_m,tr_1)\oplus AN_{FR}(st_m,tr_2)$ von tr_1 und tr_2 von st_m abgezogen werden müssten und

(3.) die Kapazität $KP(st_m)$ keiner Stelle st_m durch die Kardinalität der Netto-Markenerzeugung
$(AN_{FR}(tr_1,st_m)\oplus AN_{FR}(tr_2,st_m))\backslash(AN_{FR}(st_m,tr_1)\oplus AN_{FR}(st_m,tr_2))$
durch tr_1 und tr_2 überschritten wird.

Wenn zwar $AKT_{ON}(tr_1,\theta_1,M_r)$ und $AKT_{ON}(tr_2,\theta_2,M_r)$ gelten, aber mindestens eine der beiden weiteren Nebenläufigkeitsbedingungen verletzt ist, liegt eine konfliktionäre Aktivierung vor.

Zu beachten ist hierbei, dass die gleichen Transitionen bei unterschiedlichen Grundsubstitutionen einmal nebenläufig und ein andermal konfliktionär zueinander aktiviert sein können. Verdeutlicht wird dies anhand eines Beispiels entsprechend Abbildung 31:

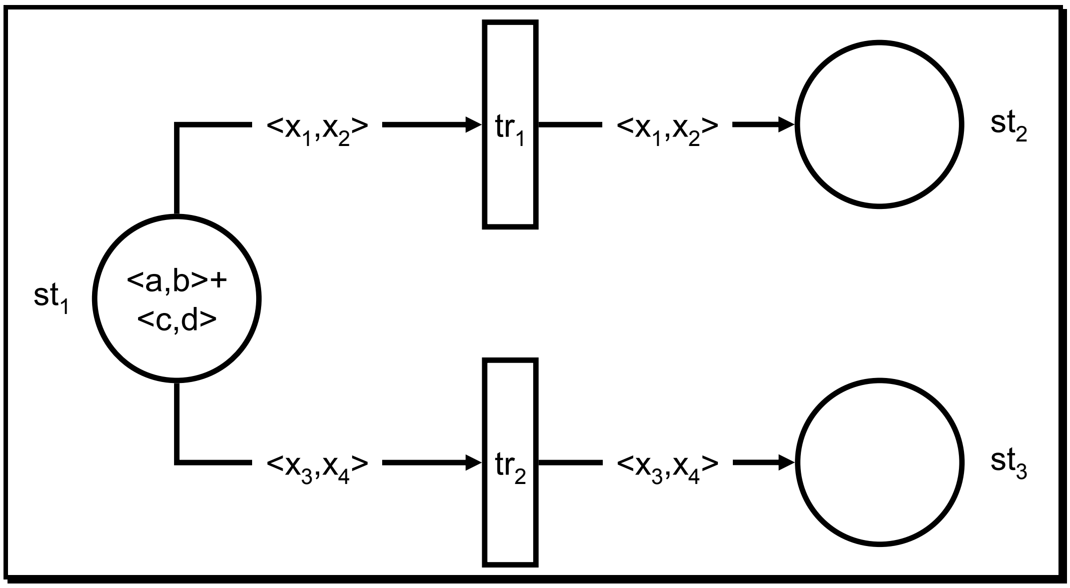

Abbildung 31: Konflikt in Ontologie-Netzen I

Unter den Grundsubstitutionen θ_{1_1} und θ_{2_1} mit

---

1) Die Bedingungen für die nebenläufige Aktivierung prozeduraler Transitionen können als Bijugat formuliert werden, weil die allgemeine Flusskantenannotationsfunktion AN_{FR} *sämtliche* Tupel $(kn_x,kn_y)\in((ST\times TR)\cup(TR\times ST)$ in ihrem Argument aufnehmen kann. Hierzu gehören auch solche Tupel (kn_x,kn_y), die nicht in der Flussrelation FR enthalten sind. Somit kann für jedes Tupel (st_m,tr_n) angegeben werden, wie hoch der Markenkonsum von der Stelle st_m durch das Schalten der Transition tr_n betragen würde, obwohl $(st_m,tr_n)\notin FR$ gilt. Die allgemeine Flusskantenannotationsfunktion AN_{FR} ordnet in diesem Fall dem Tupel (st_m,tr_n) die Annotation $\varnothing_M$ zu.

$\theta_{1_1}=\{(x_1,a),(x_2,b)\}$ und
$\theta_{2_1}=\{(x_3,a),(x_4,b)\}$

gelten zwar $AKT_{ON}(tr_1,\theta_{1_1},M_r)$ und $AKT_{ON}(tr_2,\theta_{2_1},M_r)$, allerdings sind dann tr_1 und tr_2 konfliktionär zueinander aktiviert. Beide Transitionen würden nämlich bei den entsprechenden Grundsubstitutionen die Marke <a,b> mit

$$<a,b> = <x_1,x_2>[\theta_{1_1}] = <x_3,x_4>[\theta_{2_1}]$$

von der Stelle st_1 abziehen. Die Multiplizität der Marke <a,b> beträgt allerdings auf der Stelle st_1 genau 1.

Unter den Grundsubstitutionen

$\theta_{1_2}=\{(x_1,a),(x_2,b)\}$ und
$\theta_{2_2}=\{(x_3,c),(x_4,d)\}$

wären die beiden Transitionen hingegen nebenläufig aktiviert, da durch das Schalten von tr_1 und tr_2 unter θ_{1_2} bzw. θ_{2_2} die Marken <a,b> mit $<a,b> = <x_1,x_2>[\theta_{1_2}]$ und <c,d> mit $<c,d> = <x_3,x_4>[\theta_{2_2}]$ von st_1 abgezogen würden.

Bezüglich schleifenartiger Ontologie-Netze kann die Bestimmung der Nebenläufigkeit aus S/T-Netzen analog übertragen werden. Auch hierbei müssen allerdings die Grundsubstitutionen in Betracht gezogen werden, um bestimmen zu können, ob ein Konflikt vorliegt.

Beispielsweise gelten für die beiden Transitionen tr_1 und tr_2 aus dem Ontologie-Netz in Abbildung 32 $AKT_{ON}(tr_1,\theta_1,M_r)$ und $AKT_{ON}(tr_2,\theta_{2_1},M_r)$ mit $\theta_1=\{(x_1,a),(x_2,b)\}$ und $\theta_{2_1}=\{(x_3,a),(x_4,b)\}$.

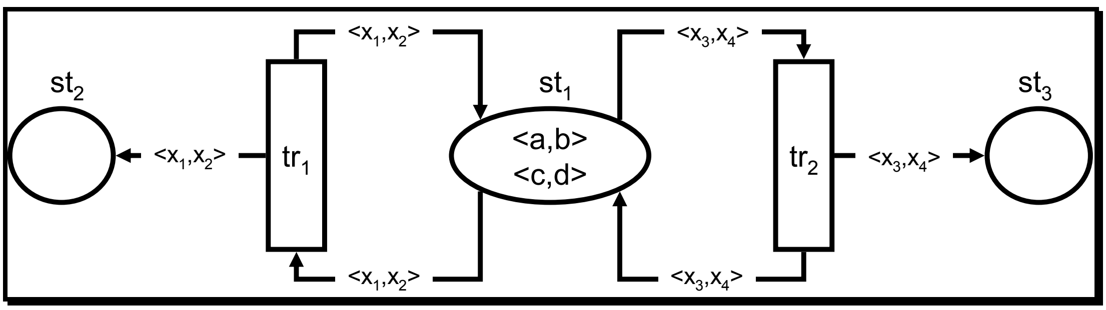

Abbildung 32: Schleife aus Ontologie-Netz

Durch das Schalten von tr_1 oder tr_2 unter θ_1 bzw. θ_{2_1} wird die Aktivierung der jeweils anderen Transition nicht aufgehoben. Dennoch sind tr_1 und tr_2 unter θ_1 bzw. θ_{2_1} konfliktionär aktiviert, weil die Stelle st_1 nicht genügend Kopien der Marke <a,b> umfasst, um sowohl tr_1 als auch tr_2 zu versorgen. Wenn hingegen statt θ_{2_1} für tr_2 die Grundsubstitution θ_{2_2} mit $\theta_{2_2}=\{(x_3,c),(x_4,d)\}$ verwendet wird, dann sind tr_1 und tr_2 unter θ_1 bzw. θ_{2_2} nebenläufig aktiviert.

Das zweite Beispiel konfliktionär aktivierter Transitionen tr_1 und tr_2 unter den Grundsubstitutionen θ_1 bzw. θ_2 wird anhand des Ontologie-Teilnetzes aus Abbildung 33 diskutiert. Im Gegensatz zum vorherigen Beispiel ist hierbei für den Konflikt nicht ausschlaggebend, dass nicht genügend Markenkopien für den Konsum der beiden Transitionen vorliegen. Ausschlaggebend bei diesem zweiten Beispiel ist die Kapazität der gemeinsamen Ausgangsstelle beider Transitionen.

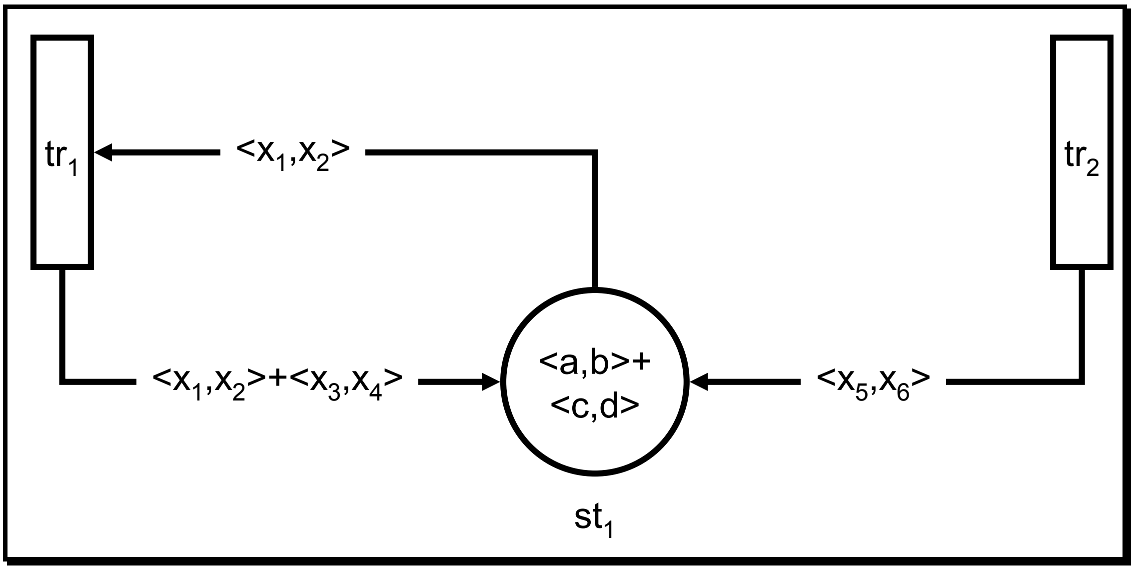

Abbildung 33: Konflikt in Ontologie-Netz II

Für die beiden Transitionen tr_1 und tr_2 gelten $AKT_{ON}(tr_1,\theta_1,M_r)$ bzw. $AKT_{ON}(tr_2,\theta_2,M_r)$ bei

$\theta_1=\{(x_1,a),(x_2,b),(x_3,e),(x_4,f)\}$,
$\theta_2=\{(x_5,e),(x_6,f)\}$ und
$KP(st_1)=3$.

Durch das alleinige Schalten von tr_1 unter θ_1 würde einerseits von st_1 die Marke <a,b> abgezogen, und es würden andererseits auf st_1 die Marken <a,b> und <e,f> abgelegt werden. Dadurch würde die Kapazität $KP(st_1)$ ausgeschöpft werden. Bei alleiniger Schaltung von tr_2 unter θ_2 würde auch die Kapazität $KP(st_1)$ ausgeschöpft werden. Jedoch können tr_1 und tr_2 nicht gemeinsam unter $\theta_{1,}$ bzw. θ_2 schalten, da hierdurch $KP(st_1)$ überschritten würde. Also sind die Transitionen tr_1 und tr_2 konfliktionär aktiviert.

Für Transitionen aus Ontologie-Netzen wird auch die Möglichkeit ihrer nebenläufigen Aktivierung zu sich selbst eingeräumt. Eine Transition tr_n ist genau dann nebenläufig zu sich selbst unter zwei Grundsubstitutionen θ_1 und θ_2 aktiviert, wenn sie sowohl unter θ_1 und als auch unter θ_2 schalten kann und keiner dieser beiden Schaltakte die Aktivierung der Transition tr_n bezüglich der jeweils anderen Grundsubstitution aufhebt. Beispielsweise ist die Transition tr_n aus der Abbildung 34 bezüglich der Grundsubstitution

$\theta_1=\{(x_1,a),(x_2,b)\}$ und
$\theta_2=\{(x_1,c),(x_2,d)\}$,

nebenläufig zu sich selbst aktiviert.

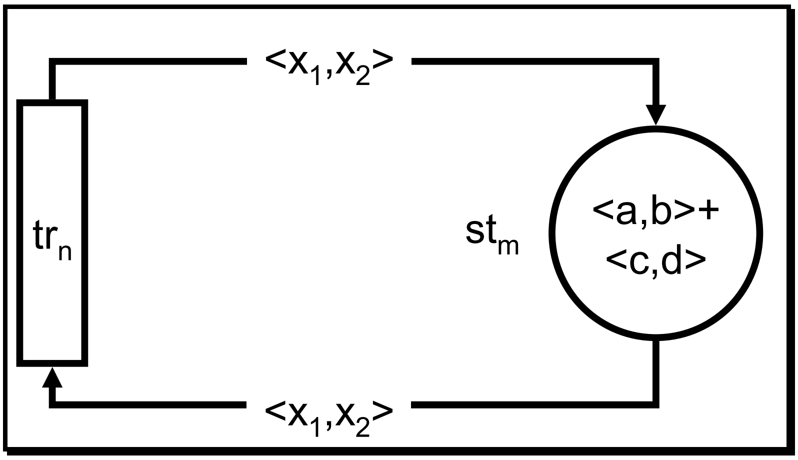

Abbildung 34: Transitionen mit selbstbezogener nebenläufiger Aktivierung

5 Evaluation von Ontologie-Netzen

5.1 Evaluation der statischen Struktur

5.1.1 Ontologie-Sprachen

5.1.1.1 Überblick über Ontologie-Sprachen

Für die Konstruktion von Ontologien wurden in den letzten Jahren mehrere Sprachen[1)] vorgestellt. Teilweise sind die Sprachen als Spielarten „traditioneller" Sprachen der Wissensrepräsentation entstanden. Andere Sprachen wurden wiederum gezielt mit der Intention formuliert, Ausdrucksmittel für die Konstruktion von Ontologien zur Verfügung zu stellen. Zu den bekanntesten Ontologie-Sprachen gehören *Ontolingua*, *F-Logic*, *SHOE*, *RDF(S)*, *DAML+OIL* und *OWL*. Ontolingua und F-Logic haben ihren Hintergrund jeweils in der prädikatenlogischen Wissensrepräsentation. Die weiteren Sprachen sind hingegen größtenteils unter dem Einfluss von Grammatiken für Auszeichnungssprachen, wie z.B. XML, entstanden.[2)] Während Ontolingua und F-Logic jeweils ihren Hintergrund im *Knowledge-Engineering* haben, zählen SHOE, RDF(S), DAML+OIL und OWL zu den Forschungsarbeiten im Rahmen des *Semantic Web*. Entsprechend ihrer Orientierung am Semantic Web sind die letztgenannten Sprachen vollkommen mit der Standard Generalized Markup Language (SGML) vereinbar. Dabei ist SHOE an der SGML-Instanz HTML ausgerichtet. Die weiteren Sprachen bedienen sich hingegen der XML-Grammatik.

Weitere Sprachen, wie z.B. CyCL[3)], OCML[4)] und XOL[5)], haben für die aktuelle Diskussion um Sprachen zur Konstruktion von Ontologien geringere Bedeutung, da sie entweder – wie im Fall von CyCL und OCML – von einer spezifischen Software abhängen oder – wie im Fall von XOL – keine umfassende Sprachspezifikation vorliegt. Daher wird im Folgenden lediglich auf die erstgenannten Sprachen näher eingegangen.

In den folgenden Abschnitten werden die Sprachen jeweils vorgestellt und hinsichtlich der zuvor aufgestellten Kriterien beurteilt. Um einen ganzheitlichen Vergleich zwischen

---

1) Im Folgenden werden Sprachen zur Konstruktion von Ontologien auch verkürzt als „Ontologie-Sprachen" bezeichnet. Zu beachten ist dabei, dass es sich – bei Beibehaltung der bisherigen Diktion – um Metasprachen handelt, die für die Spezifikation objektsprachlicher Konstrukte verwendet werden können.

2) Zudem weisen DAML+OIL und OWL Bezüge zu *terminologischen Logiken* auf, die wiederum auf die Prädikatenlogik zurückgeführt werden können. Auf diese Aspekte von DAML+OIL und OWL wird in den entsprechenden Abschnitten 5.1.1.6 bzw. 5.1.1.7 eingegangen.

3) Vgl. LENAT ET AL. (1990), S. 33 ff.; LENAT (1995), S. 45 ff.

4) Vgl. DOMINGUE ET AL. (1999), S. 23 ff.

5) Vgl. KARP ET AL. (1999)

Ontologie-Sprachen durchführen zu können, wird der Kriterienkatalog für die *statische* Struktur herangezogen, der in Abschnitt 2.1.3.2.2 vorgestellt wurde. Mit Hilfe des Kriterienkataloges ist es möglich, die unterschiedlichen Ontologie-Sprachen miteinander systematisch zu vergleichen. Darüber hinaus ist ein solcher Kriterienkatalog eine notwendige Voraussetzung, um Ontologien evaluieren zu können. Jede Ontologie wird nämlich mittels einer Ontologie-Sprache konstruiert. Um die Güte einer Ontologie beurteilen zu können, muss zunächst die Güte der Ontologie-Sprache bestimmt werden.[1] Schließlich wird eine Referenzontologie in jeder der genannten Sprachen rekonstruiert, um einen näheren Eindruck der Sprachen zu vermitteln.

5.1.1.2 Ontolingua

Die Ontologie-Sprache *Ontolingua*[2] ist am *Knowledge Systems Laboratory*[3] der Universität Stanford entwickelt worden. Für die Entwicklung von Ontolingua wurde auf den Arbeiten zur Wissensrepräsentationssprache *Knowledge Interchange Format* (KIF)[4] aufgesetzt. Während KIF in erster Linie die Funktion einer „Medial-Sprache" zwischen verschiedenen – nicht notwendigerweise ontologiegestützt konstruierten – wissensbasierten Systemen übernimmt, ist Ontolingua explizit auf die Konstruktion von Ontologien ausgerichtet. Für diesen Zweck wurde KIF um Ausdrucksmittel erweitert, die für die Konstruktion von Ontologien nötig sind. Es handelt es sich hierbei um Auszeichnungsmöglichkeiten sowohl für objektsprachliche Konstrukte, wie z.B. Konzepte und Relationen, als auch für metasprachliche Konstrukte, wie z.B. Ontologie-Bezeichnungen, Anzahl der Konzepte usw. Anhand dieser zusätzlichen Konstrukte ermöglicht Ontolingua beispielsweise die Referenzierung einer Ontologie in einer anderen Ontologie.[5] Dadurch können in einer zweiten Ontologie sprachliche Konstrukte wiederverwendet werden, deren Semantik bereits in einer ersten Ontologie spezifiziert wurde. Die Auflösung eventueller Konflikte zwischen den Spezifikationen wird allerdings auf die jeweilige Implementierung verlagert. Ontolingua selbst stellt hingegen keine Ausdrucksmittel für die Auflösung von Konflikten bereit.

KIF und Ontolingua basieren beide auf Prinzipien der prädikatenlogischen Wissensrepräsentation.[6] Die Notation von Ausdrücken unterscheidet sich im Wesentlichen von der konventionellen Prädikatenlogik dadurch, dass logische Symbole einem maschinenverarbeitbaren Zeichensatz entnommen und in Pre-fix-Notation angegeben werden. Dabei wird in einer Ontolingua-Ontologie stets Bezug auf die so genannte *Frame-*

---

1) Vgl. FRANK (2000), S. 346, mit Bezug auf konzeptionelle Modelle: „Die Evaluation von konzeptionellen Modellen setzt also die Beurteilung der verwendeten Modellierungssprache voraus.".

2) Vgl. GRUBER (1992), S. 5 ff.; GRUBER (1993), S. 202 ff.; FARQUHAR ET AL. (1997), S. 710 ff.

3) http://www.ksl.stanford.edu/ .

4) Vgl. GENESERETH/FIKES (1997); SOWA (2000), S. 25 f.

5) Vgl. FARQUHAR ET AL. (1997), S. 715.

6) Vgl. FARQUHAR ET AL. (1997), S. 708; GÓMEZ-PÉREZ ET AL. (2004), S. 205; GRUBER (1992), S. 3.

Ontology[1)] genommen, die selbst in KIF spezifiziert ist. In der Frame-Ontology sind Ausdrucksmittel spezifiziert, mit denen Ontologien in Ontolingua konstruiert werden können. Somit hat die Frame-Ontology den Charakter einer „Meta-Ontologie", in der Ausdrucksmittel für ontologische Spezifikationen definiert sind. Die spezifizierten Ausdrucksmittel der eigenen Ontologie werden jeweils als Instanzen der Ausdrucksmittel in der Frame-Ontology behandelt.

Die Spezifikation von *Konzepten* erfolgt in Ontolingua nach folgendem Schema:[2)]

(define-class *Name* (?*Variable*)	
"Definition*s-String*"	
:def *oder* :iff-def	*KIF-Ausdruck*
:constraints	*KIF-Ausdruck*
:sufficient	*KIF-Ausdruck*
:default-constraints	*KIF-Ausdruck*
:equivalent	*KIF-Ausdruck*
:axioms	*KIF-Ausdruck*)

Der Platzhalter *?Variable* wird bei einer konkreten Spezifikation durch eine Variable ersetzt, deren Allquantifizierung für die folgenden KIF-Ausdrücke unterstellt wird. Über diese Variable hinaus können in den folgenden Ausdrücken weitere Variablen verwendet werden. Je nachdem, ob ein Konzept oder eine Relation spezifiziert wird, kann die Anzahl der Variablen in der ersten Zeile variieren, wobei für Konzepte lediglich eine Variable zugelassen ist. Für Relationen wird mit der Anzahl der Variablen ihre Stelligkeit angegeben.

Mit dem KIF-Ausdruck im Anschluss an *:def* werden notwendige Bedingungen für Instanzen des jeweiligen Konzepts ausgedrückt. Im Anschluss an *:iff-def* werden sowohl notwendige als auch hinreichende Bedingungen ausgedrückt. Somit gelten einerseits die Bedingungen für alle Instanzen des Konzepts. Andererseits sind alle formalen Objekte, die die entsprechenden Bedingungen erfüllen, Instanzen des Konzepts. Der KIF-Ausdruck im Anschluss an *:sufficient* entspricht hinreichenden Bedingungen, die alle Instanzen des Konzepts erfüllen müssen. Im Anschluss an *:constraints* und *:defaul-constraints* werden jeweils Konzept-Attribute bzw. Default-Werte der Konzept-Attribute spezifiziert. Mit *:equivalent* können Äquivalenzen zwischen Konzepten ausgezeichnet werden.

---

1) Vgl. GRUBER (1993), S. 204 u. 211 ff.,Neben der Frame-Ontology spielt in neueren Versionen von Ontolingua auch die *OKBC-Ontology* eine Rolle; vgl. GÓMEZ-PÉREZ ET AL. (2004), S. 48 ff. u. 205 ff.

2) In Anlehnung an GRUBER (1992), S. 5. Zu beachten ist, dass Konzepte in Ontolingua als *Klassen* (Classes) bezeichnet werden.

Mit KIF-Ausdrücken im Anschluss an *:axioms* werden weitere Eigenschaften des jeweiligen Konzepts ausgegeben. Beispielsweise wird die taxonomische Struktur von Konzepten mittels der beiden metasprachlichen Relationen *Superclass-of* und *Subclass-of* im Anschluss an den Ausdruck *:axioms* spezifiziert.[1)]

Ontolingua erlaubt die Auszeichnung von Konzepten als Instanzen von anderen Konzepten. Somit ist die Spezifikation von *Meta-Konzepten* in Ontolingua zugelassen. Darüber hinaus ist es in der Implementierung des gleichnamigen, webbasierten Software-Tools *Ontolingua*[2)] notwendig, bei jeder Ontologie-Konstruktion auf die *Frame-Ontology* zu verweisen. Die Ausdrucksmittel der eigenen Ontologie werden reifiziert und den jeweiligen Extensionen der Frame-Ontology zugewiesen. Durch diese Reifikation von Ausdrucksmitteln können ihnen jeweils auch – entsprechend den Attributen aus der Frame-Ontology – Attributswerte zugewiesen werden. Benutzerdefinierte Attribute für Konzepte sind hingegen nicht zugelassen.[3)] Zudem sind zwar mengenwertige Konzepte zugelassen, allerdings nicht in der Frame-Ontology vorgesehen. Dafür stellt KIF metasprachliche Ausdrücke, wie z.B. *set* und *individual*, zur Verfügung.

Die Spezifikation von Relationen erfolgt anhand von Schemata, die dem o.a. Schema für Konzepte ähnlich sind. Die Unterschiede liegen darin, dass anstelle des Ausdrucks *:define-class* die Ausdrücke *:define-function* und *:define-relation* verwendet werden. Zudem wird mit der Anzahl der Variablen in der ersten Spezifikationszeile die Stelligkeit des jeweiligen Symbols angegeben. Dabei können Relationen mit beliebiger Stelligkeit spezifiziert werden. Zudem können Unterordnungsbeziehungen zwischen Relationen definiert werden. Darüber hinaus sind in Ontolingua natürlichsprachliche Definitionen für objektsprachliche Konstrukte formulierbar.

Für die Spezifikation von Kardinalitäten sind in Ontolingua mehrere – teilweise redundante – Ausdrucksmöglichkeiten reserviert. Einerseits können Relationen als Instanzen von kardinalitätsspezifischen Meta-Konzepten definiert werden. Beispielweise lassen sich Relationen als Instanzen des Meta-Konzepts *Function* spezifizieren, wodurch ihre Maximumkardinalität auf 1 festgelegt wird. Andererseits können auch Minimumkardinalitäten für Relationen durch alternative Ausdrucksmittel angegeben werden.

Sowohl Inferenz- als auch Integritätsregeln lassen sich in Ontolingua konstruieren. Auch hierbei werden die logischen Symbole jeweils in einer Pre-fix-Notation verwendet. Beispielsweise kann die Regel $\forall x_1,x_2: R_1(x_1,x_2) \wedge R_2(x_1,x_2) \rightarrow R_3(x_1,x_2)$ in Ontolingua wie folgt ausgedrückt werden:

---

1) Vgl. GRUBER (1992), S. 13.

2) http://www.ksl.stanford.edu/software/ontolingua/
Falls nicht explizit hervorgehoben, wird im Folgenden auf die *Sprache* Bezug genommen, wenn die Bezeichnung *Ontolingua* verwendet wird.

3) Vgl. GRUBER (1993), S. 214.

```
(define-Axiom
:=  (forall (?x1)
        (forall (?x2)
            (=>    (and    (R1 ?x1 ?x2)
                           (R2 ?x1 ?x2))
                   (R3 ?x1 ?x2)))))
```

Darüber hinaus können in Ontolingua weder Bezeichnungen noch Definitionen für Regeln angegeben werden.

5.1.1.3 F-Logic

Die Sprache Frame-Logic (F-Logic) entstammt ursprünglich der Forschung zu dem Themengebiet objektorientierter deduktiver Datenbanken.[1)] Seit einigen Jahren wird F-Logic darüber hinaus auch für die Konstruktion von Ontologien diskutiert.[2)] Durch die Kombination von Prinzipen objektorientierter Systemstrukturierung einerseits und deduktiver Datenbanken andererseits weist F-Logic eine hohe Ausrucksstärke auf, die sich zu Zwecken der Ontologie-Konstruktion verwenden lässt. Im Unterschied zu Ontolingua existiert für F-Logic-Ontologien keine „Meta-Ontologie“, in der Ausdrucksmittel für die Spezifikation von Ontologie-Komponenten vorgesehen wären. Die Ausdrucksmittel von F-Logic sind jeweils in der F-Logic-Syntax und -Semantik „verdrahtet“.

In F-Logic können lediglich die taxonomischen Beziehungen zwischen Konzepten unmittelbar ausgedrückt werden. Als metasprachliche Ausdrucksmittel sind hierfür zwei aufeinander folgende Doppelpunkte „::“ reserviert. Der einfache Doppelpunkt „:“ ist hingegen dafür reserviert, die Zugehörigkeit eines formalen Objekts zur Extension eines Konzepts auszuzeichnen. Um Inkompatibilität und Äquivalenz von Konzepten ausdrücken zu können, müssen in F-Logic entsprechende Integritäts- oder Inferenzregeln konstruiert werden (s.u.).

Meta-Konzepte können in F-Logic unmittelbar spezifiziert werden, da Konzepte und ihre Instanzen jeweils dem gleichen Individuenbereich zugezählt werden.[3)] Somit können Konzepten auch Attributswerte entsprechend den Attributen der jeweiligen Meta-Konzepte zugewiesen werden. Beispielsweise ist die folgende Spezifikation in F-Logic zulässig:

---

1) Vgl. LAUSEN/VOSSEN (1996), S. 216 ff.

2) Vgl. ANGELE/LAUSEN (2004), S. 29 ff.

3) Vgl. KIFER ET AL. (1995), S. 747.

$Konzept_1$	::	$Konzept_2$.
$Objekt_1$	:	$Konzept_1$.
$Objekt_2$	:	$Konzept_3$.
$Konzept_3$	:	$Konzept_1$.

Das Konzept *$Konzept_1$* wird in der Beispielspezifikation dem Konzept *$Konzept_2$* durch „::" taxonomisch untergeordnet. Die formalen Objekte *$Objekt_1$* und *$Objekt_2$* werden durch „:" als Instanzen der Konzepte *$Konzept_1$* bzw. *$Konzept_3$* spezifiziert. Schließlich wird das Konzept *$Konzept_3$* als Instanz des Konzepts *$Konzept_1$* spezifiziert. Entsprechend handelt es sich bei $Konzept_1$ sowohl um ein objektsprachliches Konzept als auch um ein metasprachliches Meta-Konzept.

Mengenwertige Konzepte sind in F-Logic nicht zulässig. Als Extension von Konzepten können nur einwertige formale Objekte angegeben werden. Darüber hinaus erlaubt F-Logic die Spezifikation sowohl von rechtsmehrdeutigen als auch von funktionalen Relationen.[1] Die Unterscheidung zwischen den beiden Konstruktarten erfolgt jeweils in ihrer Typisierung, wobei die Typisierung nachstehendem Schema folgt:

Argumentkonzept[	$Operation_1$	=> $Zielkonzept_1$;
	$Operation_2$	=> $Zielkonzept_2$;
	$Relation_1$	=>> $Zielkonzept_3$;
	$Relation_2$	=>> $Zielkonzept_4$].

Mit dem Ausdrucksmittel „=>" werden funktionale Relationen – also Operationen spezifiziert. Beispielsweise werden in der oben angegebenen Spezifikation die Operationen *$Operation_1$* und *$Operation_2$* an erster Stelle mit dem Konzept *Argumentkonzept* und an zweiter Stelle mit *$Zielkonzept_1$* bzw. *$Zielkonzept_2$* typisiert. Mit „=>>" werden hingegen rechtsmehrdeutige Relationen spezifiziert. In der Beispielspezifikation werden die Relationen *$Relation_1$* und *$Relation_2$* mit an erster Stelle mit dem Konzept *Argumentkonzept* und an zweiter Stelle mit den Konzepten *$Zielkonzept_3$* bzw. *$Zielkonzept_4$* typisiert.

Sowohl „=>" als auch „=>>" entsprechen in der formalen F-Logic-Semantik *partiellen* Relationen.[2] Insofern entspricht „=>" die Min-Max-Kardinalität (0;1) und „=>>" die Min-Max-Kardinalität (0;n).

Um Formeln auf der Instanzenebene in F-Logic konstruieren zu können, werden die Zeichen => und =>> durch -> bzw. ->> ersetzt. Anstelle der Konzepte werden hierbei die Instanzen der Konzepte aufgeführt. Die oben aufgeführte Beispielspezifikation verdeutlicht bereits, dass für Relationen nur die Spezifikation jeweils eines Argument- und

1) Eine Unterscheidung zwischen Operations- und Relationssymbolen findet sich in F-Logic nicht in dieser Diktion. Anstelle dessen wird von einwertigen bzw. mengenwertigen *Attributen* oder *Methoden* gesprochen; vgl. GÓMEZ-PÉREZ ET AL. (2004), S. 233, zur Bezeichnung *Attribute* und KIFER ET AL. (1995), S. 754 f., zur Bezeichnung *Methode*.

2) Vgl. KIFER ET AL. (1995), S. 758.

eines Zielkonzepts zulässig ist. Mehrere Argument- und Zielkonzepte können in F-Logic nicht ausgedrückt werden. Somit sind n-stellige Operations- und Relationssymbole mit n>2 in F-Logic nicht zugelassen.

Für die hierarchische Anordnung von Relationen ist in F-Logic kein eigenes Ausdrucksmittel vorgesehen, das den beiden Doppelpunkten für Konzepthierarchien entspräche. Dennoch können Ordnungsbeziehungen zwischen Relationen durch entsprechende *Inferenzregeln* ausgedrückt werden. Beispielsweise wird durch folgende Regel die Relation R_1 der Relation R_2 untergeordnet:[1)]

```
FORALL X,Y:
    X [R1 ->> Y]
        → X [ R2 ->> Y].
```

Analog zu der o.a. Regel können Regeln zur Ordnung von Operationen spezifiziert werden. Darüber hinaus können durch entsprechende Inferenzregeln *Ordnungsrelationen* definiert werden. Dafür werden die jeweiligen Eigenschaften, wie z.B. Reflexivität, Symmetrie und Transitivität, als Inferenzregeln spezifiziert. Neben solchen Inferenzregeln können in F-Logic auch Integritätsregeln ausgedrückt werden. Die spezifische Verarbeitung von Integritätsregeln wird jedoch der jeweiligen Software überlassen.

Mit Hilfe von Meta-Konzepten können in F-Logic sowohl natürlichsprachliche Bezeichner als auch natürlichsprachliche Definitionen für Konzepte angegeben werden. Dies erfolgt durch die *Reifikation* von Konzepten. Bei einer Reifikation der objektsprachlichen Konzepte geht allerdings die Differenzierung zwischen Konzepten und Instanzen, über die Aussagen gemacht werden, verloren. Um der fehlenden formalen Differenzierung zwischen den beiden Konstrukttypen entgegenwirken zu können, wird im Folgenden eine beispielhafte F-Logic-Ontologie vorgestellt, die es erlaubt, objekt- und metasprachliche Konstrukte in einer Ontologie zu spezifizieren und dennoch eine Unterscheidung zwischen beiden Konstrukttypen beizubehalten.

Für die Differenzierung zwischen objekt- und metasprachlichen Konstrukten können in einer F-Logic-Ontologie die beiden metasprachlichen Konzepte *metasprachliche_Entitaet* und *objektsprachliche_Entitaet* als Subkonzepte des metapsrachlichen Maximalkonzepts *Entitaet* eingeführt werden. Dadurch ist es möglich, objektsprachliche Ausdrucksmittel in den Argumenten von metasprachlichen Ausdrucksmitteln zu verwenden. Als solche metasprachlichen Ausdrucksmittel kämen insbesondere solche Meta-Konzepte in Frage, mit denen den objektsprachlichen Konzepten metasprachliche Bezeichnungen und Definition zugewiesen werden können.

---

1) Subjugatsregeln in F-Logic werden in der vorliegenden Arbeit der prädikatenlogischen Notation folgend angegeben. So wird der Antezedenzteil einer Subjugatsformel links vom Subjunktor „→“ und der Konklusionsteil rechts vom Subjunktor notiert. Üblicherweise werden jedoch – in Anlehnung an die Schreibweise in der logischen Programmierung – Antezedenz und Konklusion in Regeln umgekehrt notiert; vgl. KIFER ET AL. (1995), S. 748.

Auf der ersten Gliederungsstufe wird das Maximalkonzept *Entitaet* der Ontologie entsprechend der folgenden Spezifikation unterteilt:

```
objektsprachliche_Entitaet   ::Entitaet.
metasprachliche_Entitaet     ::Entitaet.
```

Das metasprachliche Konzept *objektsprachliche_Entitaet* umfasst als Subkonzepte[1)] alle objektsprachlichen Konzepte, die für die Repräsentation des Objektwissens über einen Realitätsausschnitt benötigt werden. Alle Subkonzepte von objektsprachliche_Entitaet werden später als Instanzen des metasprachlichen Konzepts *metasprachliche_Entitaet* spezifiziert. Zu diesem Zweck sind dem Konzept *metasprachliche_Entitaet* solche metasprachlichen Konzepte taxonomisch untergeordnet, mit denen das Metawissen über objektsprachliche Ausdrucksmittel ausgedrückt werden kann. Die Spezifikation erfolgt folgendermaßen:

1) Bereits hier wird ein Bruch zwischen Objekt- und Metasprache deutlich. Denn die Subkonzeptrelation „::“ setzt objektsprachliche Konzepte in taxonomische Beziehungen zueinander. Bei der beispielhaften Spezifikation wird hingegen das metasprachliche Konzept *objektsprachliche_Entitaet* als Superkonzept zu objektsprachlichen Konzepten festgelegt. Es wurde allerdings darauf hingewiesen, dass in F-Logic eine Vermengung objektsprachlicher Konstrukte mit metasprachlichen Konstrukten stattfindet.

```
1.) derivative_metasprachliche_Entitaet      ::metasprachliche_Entitaet.
    originaere_metasprachliche_Entitaet      ::metasprachliche_Entitaet.

2.) Konzept            ::derivative_metasprachliche_Entitaet.
    Relation           ::derivative_metasprachliche_Entitaet.
    Operation          ::Relation.

3.) derivative_metasprachliche_Entitaet[
                       hat_Bezeichnung=>>Bezeichnung;
                       hat_Definition=>Definition].

4.) Definition       ::originaere_metasprachliche_Entitaet.
    Bezeichnung      ::originaere_metasprachliche_Entitaet.

5.) Definition[      definiert=>derivative_metasprachliche_Entitaet].
    Bezeichnung [    bezeichnet=>>derivative_metasprachliche_Entitaet;
                     synonym_zu=>>Bezeichnung;
                     homonym=>BOOLEAN].

6.) FORALL X,Y:
        X:Bezeichnung[bezeichnet->>Y] ↔
        Y:derivative_metasprachliche_Entitaet[hat_Bezeichnung->>X].

7.) FORALL X,Y:
        X:Definition[definiert->Y] ↔
        Y:derivative_metasprachliche_Entitaet[hat_Definition->X].
```

Mit den Spezifikationen 1.) und 2.) werden die Subkonzepte der Konzepte *metasprachliche_Entitaet* und *derivative_metasprachliche_Entitaet* festgelegt. Als metasprachliche Entitäten werden *Konzept*, *Relationen* und *Operationen* als Subkonzepte des Konzepts *derivative_metasprachliche_Entitaet* zugelassen. Das Konzept Operation ist hierbei dem Konzept *derivative_metasprachliche_Entitaet* mittelbar untergeordnet, da es unmittelbar dem Konzept *Relation* untergeordnet ist. Für Konzepte, Relationen und Operationen können im Anschluss an ihre Reifikation sowohl Bezeichner als auch Definitionen angegeben werden. Bei den Konzepten *Bezeichner* und *Definition* handelt es sich wiederum um originäre metasprachliche Entitäten.

Mit den Regeln 6.) und 7.) werden die inversen Beziehungen zwischen den metasprachlichen Relationen *bezeichnet* und *definiert* mit *hat_Bezeichnung* bzw. *hat_Definition* festgelegt. Eine Bezeichnung steht demnach genau dann für eine derivative metasprachliche Entität, wenn das objektsprachliche Konstrukt durch den metasprachlichen Bezeichner bezeichnet wird. Analog steht eine metasprachliche Definition genau dann für eine derivative metasprachliche Entität, wenn das jeweilige objektsprachliche Konstrukt durch sie definiert wird.

Die objektsprachlichen Instanzen der metasprachlichen Konzepte *Konzept* und *Relation* werden anhand der folgenden Inferenzregeln definiert:

```
1.) FORALL X,Y:
        X::objektsprachliche_Entitaet →
            X:Konzept.

2.) FORALL X,Y,R:
        X[R=>>Y]→
            R:Relation.
3.) FORALL X,Y,R:
        X[R=>Y]→
            R:Operation.

4.) FORALL X,Y,Z,R:
    R:transitive_Relation
    AND  X[R->>Y]
    AND  Y[R->>Z] →
                X[R->>Z].
```

Mit den ersten drei Regeln werden objektsprachliche Konzepte, Relationen und Operationen reifiziert und den Extensionen der metasprachlichen Konzepte *Konzept*, *Relation bzw. Operation* zugewiesen. Mit der ersten Regel wird das metasprachliche Konzept *Konzept* mit allen Subkonzepten des metasprachlichen Konzepts *objektsprachliche_Entitaet* instanziiert. Für die Reifikation von Relationen ist der Rückgriff auf solche Spezifikationen notwendig, in denen die Relation vorkommt. Eine Relation ist dann eine Instanz des metasprachlichen Konzepts *Relation*, wenn es eine Spezifikation gibt, in der die Relation verwendet wurde. Analog wird eine Operation durch die Regel 3.) dann der Extension des metasprachlichen Konzepts Operation zugewiesen, wenn es in der Ontologie eine Spezifikation gibt, in der die Operation verwendet wird. Mit der Regel 4.) wird schließlich exemplarisch aufgezeigt, wie die Eigenarten transitiver Relationen für alle Instanzen des metasprachlichen Konzepts *transitive_Relation* formuliert werden können.[1)]

Für die metasprachlichen Konzepte *Konzept* und *Relation* gelten folgende Spezifikationen:

1) Das setzt wiederum voraus, dass das metasprachliche Konzept *transitive_Relation* als Subkonzept des metasprachlichen Konzepts *Relation* spezifiziert wurde.

```
1.) Konzept[     Argumentkonzept_von    =>>Relation;
                 Zielkonzept_von        =>>Relation].

2.) Relation[    hat_Argumentkonzept    =>Konzept;
                 hat_Zielkonzept        =>Konzept].

4.) Operation[   hat_Argumentkonzept    =>Konzept;
                 hat_Zielkonzept        =>Konzept].

5.) FORALL X,Y,R:
        X[R=>>Y] OR X[R=>Y] →
                 R[hat_Argumenkonzept->X]
            AND  R[hat_Zielkonzept->Y]
            AND  X[ist_Argumentkonzept_von->>R]
            AND  Y[ist_Zielkonzept_von->>R].
```

Das Konzept *orginaeres_metasprachliches_Konzept* wird in die beiden Subkonzepte *Bezeichnung* und *Definition* unterteilt. Sowohl bei den Instanzen von *Bezeichnung* als auch bei den Instanzen von *Definition* handelt es sich um metasprachliche *Zeichenketten*. In der Regel sind die Instanzen von Bezeichnung *Wörter* und die Instanzen von Definitionen *Sätze*, die aus Wörtern zusammengesetzt werden. Bei den Instanzen des Konzepts *Bezeichnung* handelt es sich um solche Zeichenketten, mit denen objektsprachliche Konstrukte metasprachlich bezeichnet werden können.

Mit Hilfe der Instanzen des Konzepts *Bezeichnung* können Bezeichnerbeziehungen formal erfasst werden. Die beiden Fälle der *Synonymie* und *Homonymie* können durch die beiden folgenden Regeln erfasst:

```
1.) FORALL X,A,B:
            X[hat_Bezeichnung->>A]
    AND     X[hat_Bezeichnung->>B]
    AND     A≠B →
                A[synonym_zu->>B].

2.) FORALL X,Y,A:
            X[hat_Bezeichnung->>A]
    AND     Y[hat_Bezeichnung->>A]
    AND     X≠Y →
                A[homonym->true].
```

Mit der Regel 1.) werden *synonyme* Bezeichner erfasst. Wenn für ein objektsprachliches Konstrukt X zwei unterschiedliche metasprachliche Bezeichner A und B angegeben sind, stehen A und B in einer Synonymie-Beziehung zueinander. Der umgekehrte Fall wird durch die Regel 2.) erfasst. Wenn ein metasprachlicher Bezeichner A für zwei unterschiedliche objektsprachliche Konstrukte X und Y angegeben ist, handelt es sich bei Erstgenanntem um eine homonyme Zeichenkette.

Bei den Instanzen des Meta-Konzepts *Definition* handelt es sich um natürlichsprachliche Erläuterungen der intensionalen Semantik objektsprachlicher Konstrukte. Zu diesem Zweck verbindet das Relationssymbol *definiert* das Konzept *Definition* mit dem Konzept *derivative_metasprachliche_Entitaet*. Dabei darf jedem objektsprachlichen Konstrukt höchstens eine Definition zugeordnet werden. Umgekehrt darf eine Zeichenkette die Definition von höchstens einem objektsprachlichen Konstrukt sein.

5.1.1.4 SHOE

Die Ontologie-Sprache *Simple HTML Ontology Extensions* (SHOE) wurde an der Universität Maryland vorwiegend durch die Arbeiten von Heflin, Hendler und Luke entwickelt.[1)] SHOE ist die Erweiterung der Auszeichnungssprache HTML um Ausdrucksmittel, mit denen Web-Dokumente um eine formale Semantik angereichert werden können.[2)] Intendierte Anwendungen von SHOE sind Suchmaschinen, mit denen nicht nur eine *zeichenorientierte*, sondern auch eine *inhaltliche* Auswertung von Web-Dokumenten möglich ist. In ihrer „reinen“ Form erlaubt HTML nämlich lediglich die Spezifikation des Layouts von Web-Dokumenten.

Konzepte[3)] werden in SHOE entsprechend dem folgendem Schema spezifiziert:[4)]

<DEF-CATEGORY NAME=„k“
    [ISA =„k_2 k_3...“]
    [DESCRIPTION =„text“]
    [SHORT =„text“]>
</DEF-CATEGORY>

Hinter dem Ausdruckmittel ISA werden alle Konzepte aufgeführt, denen gegenüber das Konzept k taxonomisch untergeordnet ist. Mit der Zeichenkette hinter DESCRIPTION wird eine natürlichsprachliche Definition für k angegeben. Hinter SHORT wird eine Zeichenkette als natürlichsprachlicher Bezeichner für k angegeben. Darüber hinaus stehen keine weiteren Ausdrucksmittel für die Spezifikation von Konzepten zur Verfügung. Entsprechend können weder Meta-Konzepte noch mengenwertige Konzepte ausgedrückt werden. Um die Äquivalenz von Konzepten ausdrücken zu können, muss auf Inferenzregeln zurückgegriffen werden. Die Inkompatibilität von Konzepten kann aller-

1) Vgl. Heflin et al. (1999), S. 4 ff.; Heflin (2001), S. 47 ff.; Heflin/Hendler (2001), S. 54 ff.

2) Vgl. Gómez-Pérez et al. (2004), S. 241.

3) Konzepte werden in SHOE als *Kategorie* (Category) bezeichnet.

4) Vgl. Heflin (2001), S. 55.

dings auch nicht durch Inferenzregeln ausgedrückt werden, da für den wechselseitigen *Ausschluss* von Instanzen in den Extensionen inkompatibler Konzepte die logische Negation als Ausdrucksmittel notwendig ist, die jedoch in SHOE nicht vorgesehen ist. Entsprechend kann die Inkompatibilität von Konzepten in SHOE nicht ausgedrückt werden. Zudem können die meisten Integritätsregeln, für deren Spezifikation die Negation nötig ist, auch nicht spezifiziert werden.

In SHOE können keine funktionalen Relationen spezifiziert werden. Da auch keine Kardinalitäten in SHOE ausgedrückt werden können, können funktionale Relationen auch nicht „simuliert“ werden.

Um Beziehungen zwischen Instanzen ausdrücken zu können, stehen nur möglicherweise rechtsmehrdeutige Relationen zu Verfügung. Die Spezifikation von Relationen erfolgt nach folgendem Schema:[1]

```
<DEF-RELATION NAME=„R“
        [DESCRIPTION      =„text“]
        [SHORT            =„text“]>
    <DEF-ARG POS=(„n“ | „FROM“ | „TO“)
            [TYPE     =„typ“]
            [SHORT  =„text“]>
</DEF-RELATION>
```

Die Ausdrucksmittel DESCRIPTION und SHORT haben jeweils die gleiche Funktionalität wie bei der Spezifikation von Konzepten. Die Typisierung von möglicherweise mehrstelligen Relationen erfolgt durch die Ausdrücke im Anschluss an DEF-ARG. Hierfür kann entweder ein numerischer Wert anstelle der Variable n verwendet werden, mit dem angegeben wird, welche Argumentstelle der Relation typisiert wird. Oder es können für binäre Relationen die Ausdrücke „FROM“ und „TO“ verwendet werden. Beispielsweise kann die Relation *arbeitet_fuer*, die das Konzept *Person* mit dem Konzept *Unternehmen* verbindet, wie folgt spezifiziert werden:

```
<DEF-RELATION NAME=„arbeitet_fuer“
        [DESCRIPTION =„Relation, um Arbeitsverhältnisse zwischen Personen und Unternehmen
        auszudrücken“]
        [SHORT          =„arbeitet für“]>
    <DEF-ARG          POS=„FROM“   TYPE=„Person“
    <DEF-ARG          POS=„TO“       TYPE=„Unternehmen“>
</DEF-RELATION>
```

Schließlich können in SHOE Inferenzregeln spezifiziert werden. Ihre Spezifikation erfolgt nach folgendem Schema:

1) Vgl. HEFLIN (2001), S. 55 f.

```
<DEF-INFERENCE
        [DESCRIPTION=„text“]>
    <INF-IF>
        Antezedenz
    </INF-IF>
    <INF-THEN>
        Konzequenz
    </INF-THEN>
</DEF-INFERENCE>
```

Bei der Antezedenz und der Konsequenz der Inferenzregel wird in SHOE zwischen „Konzept-“, „Relations-“ und „Vergleichsklauseln“ unterschieden.[1)] Mit einer Konzeptklausel werden Instanzen von Konzepten abgefragt. Mit einer Relationsklausel werden die Instanzen von Relationen abgefragt. Vergleichklauseln sind insofern Relationsklauseln untergeordnet, als dass auch bei ihnen die Extensionen von Relationen abgefragt werden. Ihre Formulierung ist jedoch in der Art vereinfacht, dass die Vergleichsklauseln auf den gängigen – in der Regel numerischen – Vergleichsrelationen („kleiner“, „größer“ usw.) basieren.

5.1.1.5 RDF(S)

Die Ontologie-Sprachen *Resource Description Framework* (RDF) und *Resource Description Framework Schema* (RDFS) wurden durch das *World Wide Web Consortium* (W3C)[2)] entwickelt und standardisiert. RDF und RDFS verhalten sich insofern komplementär zueinander, als dass mit RDF grundsätzliche Möglichkeiten zur Konstruktion von Ausdrücken und mit RDFS metasprachliche Ausdrucksmittel für die Spezifikation objektsprachlicher Ausdrucksmittel zur Verfügung gestellt werden. Da für die Konstruktion von Ontologien beide Sprachen gemeinsam betrachtet werden müssen, wird im Folgenden unter der Bezeichnung *RDF(S)* auf beide Sprachen gemeinsam referenziert.

Im Zuge der Standardisierungsbemühungen des W3C ist für RDF(S) eine XML-Syntax entwickelt worden.[3)] Unabhängig davon werden auch prädikatenlogische Notationen und semantische Netze für RDF(S)-Ontologien intensiv diskutiert.[4)] Die weiteren Ausführungen orientieren sich zwar an der XML-Syntax, jedoch werden *mengentheoreti-*

---

1) Vgl. HEFLIN (2001), S. 58 ff.

2) http://www.w3.org/

3) Vgl. BECKETT/MCBRIDE (2004); HJELM (2001), S. 31 ff.; LASSILA/SWICK (1999).

4) Vgl. CORBY ET AL. (2000), S. 470 ff.; COENEN/KLAPSING (2001), S. 183 ff.; KLAPSING (2003), S. 63 ff.; MCDERMOTT/DOU (2002), S. 252 ff.

sche Aspekte dann in die Untersuchung einfließen, wenn sie für die jeweils vorgetragene Argumentation angebracht sind.

Elementar für die Konstruktion von RDF(S)-Aussagen sind die metasprachlichen Ausdrucksmittel *rdfs:Resource, rdf:Property, rdfs:ConstraintProperty, rdfs:Class* und *rdfs:Literal und rdf:Statement*. Sie werden in RDF(S) jeweils als Instanzen von *rdfs:Class* konzeptualisiert und entsprechend als *RDF(S)-Klassen* bezeichnet. Darüber hinaus stehen u.a. die metasprachlichen Relationen *rdfs:SubClassOf*, *rdf:type*, *rdfs:subPropertyOf*, *rdfs:domain* und *rdfs:range* zur Verfügung. Mit der metasprachlichen Relation *rdfs:SubClassOf* wird die Subklassen-Beziehung zwischen zwei RDF(S)-Klassen ausgedrückt. Mit *rdf:type* wird hingegen ausgedrückt, dass eine Ressource Instanz einer RDF(S)-Klasse ist. *rdfs:domain* und *rdfs:range* werden jeweils dazu verwendet, die Argumentstellen metasprachlicher Relationen zu typisieren. Da *rdfs:domain* und *rdfs:range* selbst auch metasprachliche Relationen sind, erfolgt ihre Typisierung durch sie selbst. Darüber hinaus sind alle metasprachlichen Relationen jeweils Instanzen der RDF(S)-Klasse *rdf:Property* und werden entsprechend als *RDF(S)-Properties* bezeichnet.

Der Zusammenhang zwischen den metasprachlichen Ausdrucksmitteln in RDF(S) ist in der folgenden Abbildung 35 dargestellt.

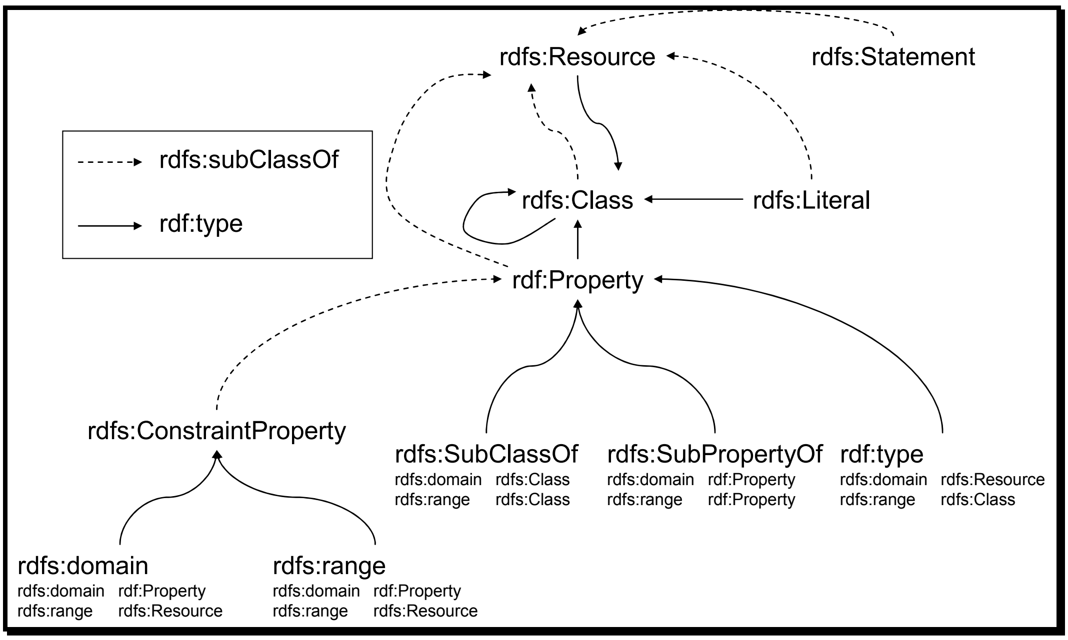

Abbildung 35: Metasprachliche Ausdrucksmittel in RDF(S)

Die Knoten in der Abbildung entsprechen RDF(S)-Klassen. Die metasprachlichen Relationen zwischen den RDF(S)-Klassen werden jeweils durch gerichtete Kanten dargestellt, wobei zwei Kantenarten unterschieden werden. Die erste Kantenart entspricht der Property *rdfs:SubClassOf*. Der Knoten am Kantenursprung wird hierdurch als Subklasse des Knotens am Kantenziel ausgezeichnet. Beispielsweise ist *rdfs:ConstraintProperty* eine Subklasse von *rdf:Property*. Die zweite Kantenart entspricht der Property *rdf:type*. Der Knoten am Kantenursprung entspricht hierbei einer Instanz des Knotens am Kantenziel. Die RDF(S)-Klasse *rdfs:SubClassOf* ist beispielsweise eine Instanz der RDF(S)-Klasse *rdf:Property*.

Schließlich ist für Properties zusätzlich noch ihre Typisierung durch *rdfs:domain* und *rdfs:range* angegeben. Beispielsweise ist die Relation *rdf:type* in ihrer ersten Argumentstelle (*rdf:domain*) mit der RDF(S)-Klasse *rdfs:Resource* und in ihrer zweiten Argumentstelle (*rdf:range*) mit der RDF(S)-Klasse *rdfs:Class* typisiert. Somit können Instanzen der RDF(S)-Klasse *rdfs:Resource* durch die Relation *rdf:type* mit Instanzen der RDF(S)-Klasse *rdfs:Class* verbunden werden.[1)]

In der RDF-Spezifikation sind *Ressourcen* definiert als: „All things being described by RDF expressions“[2)]. Somit werden alle Konstrukte, über die mittels RDF(S)-Statements Aussagen gemacht werden können, als *Ressource* bezeichnet. Der Zugriff auf Ressourcen erfolgt durch *Uniform Resource Identifiers* (URI), deren syntaktischer Aufbau durch den Standard RFC2396 der *Internet Engineering Task Force* geregelt ist.[3)] Als URI kommt dabei entweder ein *Uniform Resource Locator* (URL) oder ein *Uniform Resource Name* (URN) in Frage. Mit URLs werden Ressourcen über ihre Lokation im Internet identifiziert. Die URL einer Ressource kann demnach beispielsweise von der Adresse des Servers abhängen, auf der sie gelagert ist. URNs sind hingegen eindeutige Namen für Ressourcen, die unabhängig von der Lokation der Ressource angegeben werden können. Entsprechend der mengentheoretischen Charakterisierung von KLAPSING lässt sich die RDF(S) Klasse *rdfs:Resource* wie folgt intensional charakterisieren:[4)]

$$\textit{rdfs:Resources} = \{x \mid x \text{ ist eine URI gemäß RFC2396}\}.$$

Eine Subklasse von *rdfs:Resources* ist die Resource *rdf:Property*:[5)]

---

1) Zu beachten ist hierbei, dass keine zeitliche Präzedenzordnung festgelegt ist, der zufolge ein formales Objekt zunächst als Instanz der RDF(S)-Klasse *rdf:Resource* spezifiziert werden müsste, um im Anschluss in der zweiten Argumentstelle der Property rdf:type eingesetzt werden zu können. Wenn ein formales Objekt nämlich als Instanz der RDF(S)-Klasse *rdfs:class* spezifiziert ist, ist es auch stets eine Instanz der RDF(S)-Klasse *rdf:Resource*, da zweitgenannte eine Subklasse der erstgenannten RDF(S)-Klasse ist.

2) LASSILA/SWICK (1999).

3) Vgl. BERNERS-LEE ET AL. (1998).

4) Vgl. KLAPSING (2003), S. 40. Die mengentheoretische Charakterisierung von RDF(S)-Ausdrucksmitteln wird hier herangezogen, um den Vergleich mit den prädikatenlogischen Ausdrucksmitteln ontologischer Signaturen zu erleichtern. Um eine solche Charakterisierung einhalten zu können, ist es notwendig, RDF(S)-Klassen als Mengen und die metasprachlichen Relationen *rdfs:subClassOf* und *rdf:type* als „Teilmenge von“- ($\subseteq$) bzw. „Element von“-Beziehung ($\in$) aufzufassen.

5) Vgl. KLAPSING (2003), S. 41. Da außerhalb der Instanzen der RDF(S)-Klasse *rdf:Property* weitere Instanzen der RDF(S)-Klasse *rdf:Resource* existieren, ist in der o.a. Formel die „echte Teilmenge von“-Beziehung ($\subset$) angegeben.

$$\textit{rdf:Property} \subset \textit{rdfs:Resources}.$$

Die Instanzen von *rdf:Property* werden einerseits dazu verwendet, Aussagen zu konstruieren. Zu diesem Zweck werden jeweils zwei Instanzen der von *rdfs:Resources* durch eine Instanz von *rdf:Property* miteinander verbunden. Andererseits können die Instanzen selbst mittels einer Aussage beschrieben werden. Dafür werden in RDF(S) vordefinierte Instanzen von *rdf:Property* herangezogen. Beispielsweise gehören hierzu Aussagen, die mit Hilfe der Relationen *rdfs:domain* oder *rdfs:range* konstruiert sind. Beide Relationen verknüpfen jeweils eine Instanz von *rdf:Property* mit einer Instanz von *rdfs:Resource*.

Beachtenswert bei Instanzen von *rdf:Property* ist, dass weder für ontologiespezifische noch für vordefinierte Instanzen von *rdf:Property* Kardinalitäten definiert werden können. Es können somit keine Unter- und Obergrenzen für das Vorkommen von Instanzen in Verknüpfungen durch Instanzen von *rdf:Property* definiert werden. Entsprechend sind auch für die vordefinierten metasprachlichen Ausdrucksmittel von RDF(S) keine Kardinalitäten spezifiziert.

Für jede Instanz von *rdf:Property* muss eine *Domain* und eine *Range* definiert werden. Die Domain entspricht der ersten Argumentstelle der Property. Sie wird anhand der Instanz *rdfs:domain* von *rdf:Property* spezifiziert. Die Range entspricht der zweiten Argumentstelle der Property. Sie wird mit der Instanz *rdfs:range* von *rdf:Property* spezifiziert. Darüber hinaus können keine Argumentstellen für Instanzen von *rdf:Property* spezifiziert werden. Somit sind in RDF(S) lediglich zwei-stellige Relationen zugelassen.

Eine weitere Subklasse von *rdfs:Resource* ist *rdfs:Class*:

$$\textit{rdfs:Class} \subset \textit{rdfs:Resources}.$$

RDF(S)-Klassen sind stets Instanzen der RDF(S)-Klasse *rdfs:Class*. Sie wird daher in der RDF(S)-Spezifikation als „class of all classes" definiert. Somit ist *rdfs:Class* auch eine Instanz ihrer selbst. Dadurch ist mit der Spezifikation von *rdfs:Class* ein Problem verbunden, das in Ausarbeitungen zu RDF(S) selten thematisiert wird. Einerseits ist die RDF(S)-Klasse *rdfs:Class* der RDF(S)-Klasse *rdfs:Resource* durch *rdfs:SubClassOf* untergeordnet. Andererseits ist allerdings *rdfs:Resource* eine Instanz der RDF(S)-Klasse *rdfs:Class*. Durch die Verkettung von *rdf:SubClassOf* und *rdf:tye* wird *rdfs:Resource* auch zu einer Instanz sowohl ihrer selbst als auch ihrer Subklasse *rdfs:Class*. In der mengentheoretischen Charakterisierung ähnelt diese Situation dem *Russell-Paradoxon*.[1] Dadurch, dass „Mengen von Mengen" zugelassen werden, wird auch eine Menge X aller solcher Mengen $Y_1,...,Y_n$ zugelassen, die sich selbst nicht zum Element

1) Vgl. QUINE (1969), S. 316 ff. Vgl. darüber hinaus STEIMANN (2000), S. 26 zur Russell-Paradoxie in der Programmiersprache Smalltalk durch die Instanziierung der Klasse *Metaclass* mit sich selbst.

haben. Eine solche Menge X müsste einerseits sich selbst zum Element haben. Andererseits würde durch ihre Zugehörigkeit wiederum ihr Ausschluss gerechtfertigt werden.[1)]

Aussagen werden in RDF(S) als *Statements* konzeptualisiert, die Ressourcen untereinander mittels einer Property verknüpfen. Somit sind *Statements* geordnete Drei-Tupel bestehend aus einer Ressource, einer Property und einer Ressource. Die Komponenten eines Statements können in Anlehnung an den natürlichsprachlichen Gebrauch auch als auch als *Subjekt-Prädikat-Objekt*-Folge charakterisiert werden, wobei es sich bei dem Subjekt und dem Objekt nicht notwendigerweise um zwei unterschiedliche Ressourcen handeln muss. In der mengentheoretischen Charakterisierung lässt sich der Zusammenhang als

$$\textit{rdf:Statement} \subseteq \textit{rdfs:Resource} \times \textit{rdf:Property} \times \textit{rdfs:Resource}$$

darstellen. Bemerkenswert an Statements ist, dass sie in ihrer reifizierten Form als Instanzen der RDF(S)-Klasse *rdf:Statement* spezifiziert werden können. Somit können Statements selbst als Subjekt oder Objekt eines „höherwertigen“ Statements fungieren.[2)]

Hinsichtlich der Spezifikation von Aussagen widersprechen sich Ausarbeitungen zu RDF(S) oftmals. Teilweise wird die Sichtweise vertreten, Aussagen seien Drei-Tupel bestehend aus einer Resource, einer Property und einer Resource *oder einem Literal*.[3)] Dabei wird allerdings davon ausgegangen, die RDF(S)-Klasse *rdfs:Literal* sei nicht – wie beispielsweise in Abbildung 35 dargestellt – eine Subklasse der RDF(S)-Klasse *rdfs:Resource*. Vielmehr werden alle Zeichenketten als Instanzen der RDF(S)-Klasse *rdfs:Literal* bezeichnet, die mit der XML-Spezifikation[4)] konform sind. Demnach würden die RDF(S)-Klasse *rdfs:Literal* wie folgt intensional charakterisiert werden:[5)]

$$\textit{rdfs:Literal} = \{x \mid x \text{ ist ein String gemäß der XML-Spezifikation}\}.$$

Bei dieser Sichtweise können mit Hilfe der Instanzen von *rdfs:Literal* Aussagen über Instanzen der RDF(S)-Klasse *rdfs:Resources* getätigt werden. Die Instanzen der RDF(S)-Klasse *rdfs:Literal* wären jedoch keine Instanzen der RDF(S)-Klasse *rdfs:Resource*. Demnach könnten keine Aussagen über Instanzen von *rdfs:Literal* kon-

---

1) U.a. aus diesem Grund wurde es in der vorliegenden Arbeit bevorzugt, von Mengenfamilien zu sprechen, wenn Mengen von Mengen gemeint waren.

2) Vgl. MANOLA/MILLER (2004), Abschnitt 4.3; BRICKLEY/GUHA (2004), Abschnitt 5.3.1.

3) Vgl. KLAPSING (2002), S. 41.

4) Vgl. BRAY ET AL. (2004).

5) Vgl. KLAPSING (2003), S. 41.

struiert werden.[1)] Dadurch kann die o.a. mengentheoretische Charakterisierung in modifizierter Form angegeben werden als:[2)]

$$rdf{:}Statement \subseteq rdfs{:}Resource \times rdf{:}Property \times (rdfs{:}Resource \cup rdfs{:}Literal).$$

Diese Sichtweise steht allerdings im Widerspruch zu der formalen RDF(S)-Spezifikation. Dort wird die RDF(S)-Klasse *rdfs:Literal* explizit als Subklasse der RDF(S)-Klasse *rdfs:Resource* spezifiziert.[3)] Dieser formalen RDF(S)-Spezifikation wird im Folgenden der Vorrang gegenüber den natürlichsprachlichen RDF(S)-Ausarbeitungen eingeräumt.

Die Spezifikation eines Konzepts k_1 kann in RDF(S) nach folgendem Schema erfolgen:

```
<rdf:Description rdf:ID='k1'>
    <rdf:type rdfs:Resource='http://www.w3.org/2000/01/rdf-schema#Class'/>
    <rdfs:SubClassOf rdfs:Resource='k2'/>
    <rdfs:Label xml:lang='lan'>text</rdfs:Label>
    <rdfs:Comment xml:lang='lan'>text</rdfs:Comment>
</rdf:Description>
```

Die Spezifikation umfasst vier Drei-Tupel im o.a. Sinn. Mit dem ersten Drei-Tupel wird das Subjekt k_1 mit Hilfe des Prädikats *rdf:type* auf das Objekt *rdfs:Class* verwiesen. Das Konzept k_1 wird durch diese Aussage als Instanz der RDF(S)-Klasse *rdfs:Class* spezifiziert. Demnach werden Konzepte als RDF(S)-Klassen konzeptualisiert. Dadurch werden objekt- und metasprachliche Ausdrucksmittel zu *einem* Individuenbereich gezählt. Die Instanzen der RDF(S)-Klasse *rdfs:Class* können entweder objektsprachliche Ausdrucksmittel – nämlich Konzepte – oder metasprachliche Ausdrucksmittel sein. Als metasprachliche Ausdrucksmittel kommen beispielsweise die bereits oben vorgestellten RDF(S)-Klassen *rdf:Property* und *rdfs:Literal* in Frage.

Mit Hilfe der Property *rdfs:SubClassOf* wird angegeben, welchem Konzept k_2 gegenüber das Konzept k_1 untergeordnet ist. Für ein Konzept können beliebig viele Konzepte als Superkonzepte angegeben werden. Somit erlaubt es RDF(S), mehrere Superkonzepte zu einem Konzept anzugeben. Für jedes Unterordnungsverhältnis muss hierfür jeweils ein neues Statement mit *rdfs:SubClassOf* als Prädikat eingeführt werden. Andererseits müssen Konzepte nicht in eine Taxonomie eingebettet sein, da für *rdfs:SubClassOf* kei-

---

1) Auf diesen Aspekt wurde bereits früher – im Kontext der Typisierung von Operations- und Relationssymbolen aus ontologischen Signaturen – hingewiesen.

2) Vgl. KLASPSING (2003), S. 41, mit der Abweichung, dass Aussagen als
$$rdf{:}Statement \subseteq rdfs{:}Property \times rdfs{:}Resource \times (rdfs{:}Resource \cup rdfs{:}Literal)$$
konzeptualisiert werden. Allerdings müsste bei einer solchen Konzeptualisierung von einem Prädikat-Subjekt-Objekt-Tripel gesprochen werden, was der intuitiven Aussagenbeschreibung zuwider läuft.

3) Vgl. BRICKLEY/GUHA (2004), Abschnitt 2.3.

ne Minimum-Kadinalität spezifiziert ist. Somit können in einer RDF(S)-Ontologie Konzepte spezifiziert werden, die zu keinem anderen Konzept in einem Unterordnungsverhältnis stehen. Darüber hinaus stellt RDF(S) allerdings keine Ausdrucksmittel zur Verfügung, mit denen die Äquivalenz oder die Inkompatibilität von Konzepten ausgedrückt werden könnte.

Dadurch, dass in RDF(S) keine Differenzierung zwischen objekt- und metasprachlichen Ausdrucksmitteln erfolgt, können Konzept-Attribute und Meta-Konzepte ohne weiteres spezifiziert werden. Für die Spezifikation von Konzept-Attributen werden metasprachliche Relationen benötigt, die auch als Instanzen der RDF(S)-Klasse *rdfs:Poperty* spezifiziert werden müssen. Dadurch können Konzepte in metasprachliche Relation zu anderen Instanzen der RDF(S)-Klasse *rdfs:Resource* stehen. Analog hierzu werden Meta-Konzepte spezifiziert. Ein Konzept wird in RDF(S) genau dann zu einem Meta-Konzept, wenn mindestens ein Konzept – also eine Instanz der RDF(S)-Klasse *rdfs:Class* – existiert, das in einer *rdf:type*-Beziehung zu dem Konzept steht.

Mit den beiden Properties *rdfs:Label* und *rdfs:Comment* werden natürlichsprachliche Bezeichnungen bzw. Definitionen zu dem Konzept angegeben. Mit Hilfe des XML-Attributs *xml:lang* kann bestimmt werden, in welcher natürlichen Sprache die Bezeichnung bzw. Definition angegeben wird. Auch hierbei gilt, dass weder eine Minimum- noch eine Maximum-Kardinalität spezifiziert sind. Somit müssen nicht für alle objektsprachlichen Konstrukte[1)] sprachspezifische Bezeichnungen oder Definitionen spezifiziert werden. Zudem können beliebig viele Zeichenketten als sprachspezifische Bezeichner oder Definitionen für ein objektsprachliches Konstrukt angegeben werden.

Für die Spezifikation mengenwertiger Konzepte sind in RDF(S) die RDF(S)-Klassen *rdfs:Container*, *rdf:Bag*, *rdf:Seq* und *rdf:Alt* vorgesehen.[2)] Der Zugriff auf die einzelnen Elemente der Instanzen mengenwertiger Konzepte erfolgt durch die Properties *rdfs:Member*, *rdf:_1*, *rdf:_2* usw. Mit *rdfs:Member* wird eine ungeordnete Zugehörigkeit zu einer Menge ausgedrückt. Mit *rdf:_1*, *rdf:_2* usw. wird die geordnete Zugehörigkeit zu einer Menge ausgedrückt. Der Zusammenhang zwischen diesen metasprachlichen Konstrukten ist in der folgenden Abbildung illustriert.

1) Wie weiter unten zu sehen sein wird, können in RDF(S) nicht nur für Konzepte, sondern auch für Relationen Bezeichnungen und Definitionen spezifiziert werden.

2) Vgl. MANOLA/MILLER (2004), Abschnitt 4.

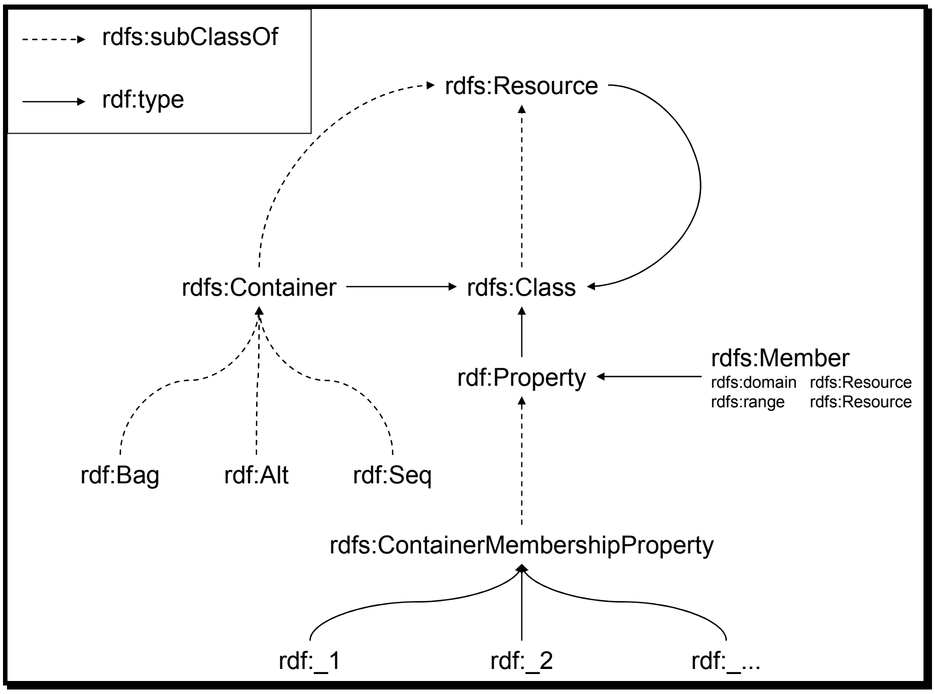

Abbildung 36: Mengenwertige Konzepte in RDF(S)

Mit Instanzen der RDF(S)-Klasse *rdfs:Container* können *Sammlungen* (Collections) von Ressourcen ausgedrückt werden. Sammlungen können als Mengen aufgefasst werden, in denen ein Element auch mehrfach vorkommen darf. Entsprechend stimmen Sammlungen mit *Multimengen*[1)] überein. Dabei werden drei unterschiedliche Sammlungsarten unterschieden. Die erste Sammlungsart wird durch die RDF(S)-Klasse *rdf:Bag* repräsentiert. Instanzen von *rdf:Bag* sind Sammlungen, in denen die einzelnen Elemente nicht in einem Ordnungsverhältnis zueinander stehen. Ein Ordnungsverhältnis muss hingegen für Instanzen der RDF(S)-Klasse *rdf:Seq* definiert werden. Bei der RDF(S)-Klasse *rdf:Alt* handelt es sich um eine einelementige Menge. Es werden jedoch mehrere Alternativen zur Verfügung gestellt, bei denen der Benutzer jeweils zu entscheiden hat, welche der Alternativen das einzige Element sein soll.

Die Spezifikation von Relationen erfolgt in RDF(S) nach folgendem Schema:

1) Vgl. Abschnitt 2.2.2.

```
<rdf:Description rdf:ID='R1'>
    <rdf:type
    rdfs:Resource='http://www.w3.org/1999/02/22-rdf-syntax-ns#Property'/>
    <rdfs:domain rdfs:Resource=k1/>
    <rdfs:range rdfs:Resource=k2/>
    <rdfs:SubPropertyOf rdfs:Resource='R2'/>
    <rdfs:Label xml:lang='ger'>text</rdfs:Label>
    <rdfs:Comment xml:lang='lan'>text</rdfs:Comment>
</rdf:Description>
```

Relationen werden in RDF(S) als Instanzen der RDF(S)-Klasse *rdf:Property* eingeführt. Somit werden Relationen in RDF(S) als *Properties* konzeptualisiert. Ihre Typisierung erfolgt u.a. mit Hilfe der Properties *rdfs:domain* und *rdfs:range*. Bei der Typisierung werden mit *rdfs:domain* und *rdfs:range* das Argument- bzw. das Zielkonzept der jeweiligen Relation angegeben. Darüber hinaus können zwischen Properties mit Hilfe der Instanz *rdfs:SubClassOf* der RDF(S)-Klasse *rdfs:SubPropertyOf* Unterordnungsbeziehungen ausgedrückt werden. Dabei ist die Property *rdfs:SubPropertyOf* sowohl in ihrer ersten als auch in ihrer zweiten Argumentstelle mit der RDF(S)-Klasse *rdf:Property* typisiert.

Properties können – wie bereits oben angesprochen – lediglich in ihrer ersten und zweiten Argumentstelle mit Hilfe der Properties *rdfs:domain* und *rdfs:range* typisiert werden. Entsprechend können nur zweistellige Relationen zwischen Konzepten ausgedrückt werden. Darüber hinaus ist es nicht möglich, die Rechtseindeutigkeit von Relationen einzufordern. Entsprechend können in RDF(S) keine funktionalen Relationen ausgedrückt werden.

Sämtliche Relationen sind Instanzen der RDF(S)-Klasse *rdfs:Resource*. Demnach können sie in den ersten Argumentstellen der Properties *rdfs:Label* und *rdfs:Comment* verwendet werden, mit denen natürlichsprachliche Bezeichner bzw. Definition für Ressourcen angegeben werden können. Jedoch sind in RDF(S) keine Ausdrucksmittel für die Konstruktion von Inferenz- oder Integritätsregeln vorgesehen. Entsprechend sind weder Ausdrucksmittel für die Bezeichnung noch Ausdrucksmittel für die Definition von Regeln vorhanden.

5.1.1.6 DAML+OIL

Die Ontologie-Sprache DAML+OIL ist im Kontext des Projektes *DARPA Agent Markup Language* (DAML) des US-amerikanischen Verteidigungsministeriums entwickelt worden. DAML+OIL ist als Erweiterung der Ontologie-Sprachen RDF(S) und *Ontology Interchange Language* (OIL)[1] konzipiert. Sämtliche Ausdrucksmittel aus RDF(S) sind

1) Das Akronym OIL wird alternativ auch als *Ontology Inference Layer* interpretiert.

auch in DAML+OIL verfügbar. Demnach kann die Menge der Ausdrucksmittel von DAML+OIL als Obermenge der Ausdrucksmittel von RDF(S) aufgefasst werden. Insofern ist die Ausdrucksmächtigkeit von DAML+OIL mindestens so groß wie die von RDF(S). Zusätzlich zu den Ausdrucksmitteln kommen Ausdrucksmittel, die ihren Ursprung in der Ontologie-Sprache OIL haben.

In DAML+OIL werden metasprachliche Ausdrucksmittel zur Spezifikation objektsprachlicher Ausdrucksmittel zur Verfügung gestellt. Daneben werden durch DAML+OIL zwei objektsprachliche Ausdrucksmittel als vordefinierte Konzepte vorausgesetzt. Es handelt sich hierbei um die Konzepte *daml:Thing* und *daml:Nothing*. Während *daml:thing* dem Maximalkonzept $\top$ in jeder DAML+OIL-Ontologie entspricht, dessen Extension alle weiteren konzeptspezifischen Extensionen umfasst, kann *daml:Nothing*, das dem Minimalkonzept $\perp$ entspricht, nicht instanziiert werden. Alle Konzepte in einer DAML+OIL-Ontologie sind Subkonzepte von *daml:Thing* und Superkonzepte von *daml:Nothing*.

Die Spezifikation eines Konzepts k erfolgt in DAML+OIL nach folgendem Schema:

```
<daml:Class rdf:ID="k">
    <daml:equivalentTo> Konzept </daml:equivalentTo>
    <daml:sameClassAs> Konzept </daml:sameClassAs>
    <daml:disjointWith> Konzept </daml:disjointWith>
    <daml:disjointUnionOf> Konzeptliste </daml:disjointUnionOf>
    <daml:unionOf> Konzeptliste </daml:unionOf>
    <daml:intersectionOf> Konzeptliste </daml:intersectionOf>
    <daml:complementOf> Konzept </daml:complementOf>
    <daml:Restriction>
        <daml:onProperty> Property </daml:onProperty>
        Restriktionen
    <daml:/Restriction>
</daml:Class>
```

Konzepte werden in DAML+OIL als Instanzen entweder von *daml:Class* oder von *daml:Datatype* spezifiziert, wobei beide eine Subklasse der RDF(S)-Klasse *rdfs:Class* sind. Somit ist jede Instanz von *daml:Class* oder *daml:Datatype* auch eine Instanz von *rdfs:Class*.[1)] Instanzen von *daml:Class* werden im Folgenden als *DAML+OIL-Klasse* bezeichnet. Instanzen von *daml:Datatype* werden hingegen als *DAML+OIL-Datentyp* bezeichnet. Gleichzeitig sind *daml:Class* und *daml:Datatype* selbst auch Instanz von *rdfs:Class*.

---

1) Dadurch, dass sowohl *daml:Class* als auch *daml:Datatype* der RDF(S)-Klasse *rdfs:Class* untergeordnet sind, muss es sich nicht notwendigerweise bei jeder RDF(S)-Klasse auch um eine Instanz von *daml:Class* oder *daml:Datatype* handeln.

Taxonomische Beziehungen zwischen DAML+OIL-Klassen können weiterhin durch *rdfs:SubclassOf* spezifiziert werden. Mit den neu hinzugekommenen Ausdrucksmitteln *daml:equivalentTo* und *daml:sameClassAs* werden Äquivalenzen von Konzepten ausgedrückt. Die DAML+OIL-Spezifikation macht in ihrer Anwendung keine Unterschiede zwischen den beiden Ausdrucksmitteln.[1)] Im Gegensatz zu *daml:equivalentTo* ist *daml:sameClassAs* eine Subrelation von *rdfs:SubclassOf*. Das bedeutet, dass, wenn zwei Konzepte in Relation *daml:sameClassAs* zueinander stehen, das erste Konzept auch immer ein Subkonzept des zweiten Konzepts sein muss. Über mögliche Symmetrieeigenschaften der beiden Properties werden darüber hinaus in der DAML+OIL-Spezifikation keine Aussagen gemacht. Daher bietet es sich an, *daml:equivalentTo* zu verwenden, wenn Auswirkungen auf die taxonomische Ordnung der Konzepte unerwünscht sind.

Mit *daml:disjointWith* kann die Inkompatibilität zweier Konzepte ausgedrückt werden. Dabei verknüpft *daml:disjointWith* jeweils ein Konzept mit einem weiteren Konzept. Im Gegensatz dazu verknüpft *daml:disjointUnionOf* ein Konzept mit einer Liste von Konzepten, wobei Letztgenannte als Instanz der DAML+OIL-Klasse *daml:List* spezifiziert werden muss. Wenn ein Konzept k in *daml:disjointUnionOf*-Beziehung zu einer Liste von Konzepten steht, bedeutet das zum einen, dass die Extensionen der Listenmitglieder paarweise disjunkt sein müssen. Zum anderen ist die Extension von k die Vereinigung der Extensionen der Listenmitglieder. Die explizite Spezifikation der paarweisen Disjunktheit von Extensionen der Listenmitglieder ist somit nicht notwendig, wenn ein Konzept in *daml:disjointUnionOf*-Beziehung zu der jeweiligen Liste steht. Wenn beispielsweise ein Konzept k in *daml:disjointUnionOf*-Beziehung zu einer Konzeptliste kl mit den Listenmitgliedern $k_1,...,k_n$ steht, bedeutet das zum einen, dass alle Instanzen jedes Konzepts k_i mit i=1,...,n auch Instanzen von k sind ($OB_k = \cup_{i=1,...,n} OB_{k_i}$) und zun anderen, dass die konzeptspezifischen Extensionen der Konzepte $k_1,...,k_n$ sich nicht überschneiden ($OB_{k_x} \cap OB_{k_y} = \varnothing$ für alle x,y=1,...,n mit $x \neq y$).

Wenn ein Konzept k in *daml:intersectionOf*-Beziehung zu einer Liste von Konzepten steht, umfasst die Extension von k alle Instanzen, die auch in den Extensionen aller Listenmitglieder enthalten sind. Die Extension von k entspricht somit der Schnittmenge der Extensionen der Listenmitglieder. Dem entgegengesetzt ist *daml:complementOf*. Wenn zwei Konzepte zueinander in *daml:complementOf*-Beziehung stehen, kann aus der Nichtzugehörigkeit eines formalen Objektes zu der Extension des einen Konzepts auf seine Zugehörigkeit zu der Extension des zweiten Konzepts geschlossen werden.

Eine wesentliche Anreicherung der Ausdrucksmächtigkeit von DAML+OIL erfolgt durch Spezifikationen im Anschluss an *daml:Restriction*. Dort werden Konzept- und Property-spezifische *Restriktionen* definiert. Es handelt sich dabei um Aussagen, die in allen Modellen zu der jeweiligen DAML+OIL-Ontologie gültig sein müssen. Restriktionen entsprechen somit *metasprachlichen Integritätsregeln*. Sie sind jeweils für das Konzept gültig, bei dessen Spezifikation sie angegeben werden. Auf welche Property

1) Vgl. CONNOLLY ET AL. (2001), Abschnitt 2.

die folgenden Restriktionen bezogen sind, wird im Anschluss an *daml:onProperty* definiert.

Zur Spezifikation der Restriktionen können metasprachliche Properties verwendet werden, die beispielsweise für die Spezifikation von Kardinalitäten nötig sind:[1)] Als solche stehen die Properties *daml:mincardinality*, *daml:maxcardinality* und *daml:cardinality* zur Verfügung, mit denen Konzept- und Property-spezifische Restriktionen mit nichtnegativen natürlichen Zahlen verknüpft werden. Mit *daml:mincardinality* kann ausgedrückt werden, mit wie vielen voneinander unterschiedlichen formalen Objekten jede Instanz des jeweiligen Konzepts *mindestens* verbunden sein muss. Analog kann mit *daml:maxcardinality* angegeben werden, mit wie vielen voneinander unterschiedlichen formalen Objekten jede Instanz des jeweiligen Konzepts *höchstens* verbunden sein kann. Die metasprachliche Property *daml:cardinality* ist insofern redundant, als dass ihre Funktionalität durch die gemeinsame Verwendung von *daml:mincardinality* und *daml:maxcardinality* erfüllt wird, falls Minimum- und Maximumkardinalität für dasselbe Konzept übereinstimmen. Mit *daml:cardinality* wird nämlich angegeben, mit wie vielen voneinander unterschiedlichen Instanzen jede Instanz aus der Extension des jeweiligen Konzepts *genau* verbunden sein muss. Wenn keine Minimum- oder Maximumkardinalitäten spezifiziert sind, wird für die jeweilige Property eine Minimumkardinalität von 0 und eine Maximumkardinalität von ∞ angenommen.[2)]

Kardinalitäten, die mit Hilfe von *daml:mincardinality*, *daml:maxcardinality* oder *daml:cardinality* ausgedrückt sind, werden als *unqualifizierte Kardinalitäten* bezeichnet. Bei ihrer Spezifikation wird nämlich die Typisierung derjenigen Property, für die die Kardinalität Gültigkeit besitzt, nicht berücksichtigt. *Qualifizierte Kardinalitäten* können mit Hilfe der metasprachlichen Properties *daml:mincardinalityQ*, *daml:maxcardinalityQ* und *daml:cardinalityQ* ausgedrückt werden. Sie sind analog zu den Ausdrucksmitteln für die Spezifikation unqualifizierter Kardinalitäten definiert. Im Gegensatz zu diesen nehmen allerdings die Ausdrucksmittel für die Spezifikation qualifizierter Kardinalitäten stets Bezug auf die zweite Argumentstelle der Property, für die die Kardinalität definiert wird. Beispielsweise kann mit *daml:mincardinalityQ* angegeben werden, mit wie vielen voneinander unterschiedlichen Instanzen des Konzepts, mit dem die Property in der zweiten Argumentstelle typisiert ist, jede Instanz des ersten Konzepts mindestens verbunden sein muss.[3)]

Über die Angabe von Kardinalitäten hinaus können weitere Konzept- und Property-spezifische Restriktionen in der Sektion unter *daml:Restriction* angegeben werden.

---

1) Vgl. GÓMEZ-PÉREZ ET AL. (2004), S. 267.

2) Vgl. CONNOLLY ET AL. (2001), Abschnitt 6.

3) Die Unterscheidung zwischen qualifizierten und unqualifizierten Kardinalitäten wird allerdings vom Verfasser als kritisch angesehen. Eine solche Unterscheidung macht nämlich nur dann Sinn, wenn zugelassen wird, dass Properties Instanzen auch dann miteinander verbinden können, wenn dies nicht mit der Typisierung der Property vereinbart werden kann. Dadurch wird allerdings die Sinnhaftigkeit der Typisierung von Properties grundsätzlich in Frage gestellt.

Hierunter kann z.B. kann für ein Konzept, das eine Property von einem Superkonzept „erbt", eine konzeptspezifische Restriktion formuliert werden. Beispielsweise kann bei der Spezifikation eines Konzepts *Mensch*, das die Property *hat_Mutter* vom Superkonzept *Säugetier* erbt, bestimmt werden, dass *hat_Mutter* an ihrer zweiten Argumentstelle nur Instanzen des Konzepts *weiblicher_Mensch* aufnehmen darf, wenn in der ersten Argumentstelle eine Instanz von *Mensch* vorkommt:

```
<daml:Restriction>
    <daml:onProperty rdfs:Resource="hat_Mutter"/>
    <daml:toClass rdfs:Resource="weiblicher_Mensch"/>
</daml:Restriction>
```

Die Eingrenzung der zweiten Argumentstelle von *hat_Mutter* auf Instanzen von *weiblicher_Mensch* erfolgt mittels der Property *daml:toClass*. Dadurch wird die Integrität der Konzeptinstanziierung bewahrt. Wenn zwischen zwei Instanzen eine Beziehung besteht, die mit einer Property definiert wurde, auf die die jeweilige Restriktion mittels *daml:onProperty* bezogen ist, dann muss das zweite formale Objekt zu der Extension des Konzepts gehören, mit dem die Property an der zweiten Argumentstelle typisiert ist. Dies hat für *alle* Instanzen zu gelten, zwischen denen eine entsprechende Beziehung spezifiziert ist. Wenn statt *daml:toClass daml:hasClass* verwendet wird, bedeutet dies, dass *mindestens eine* Instanz aus der Extension desjenigen Konzepts existiert, mit dem die Property in ihrer zweiten Argumentstelle typisiert ist und das durch diese Property mit einer Instanz aus der aus Extension des Konzepts, mit dem die Property in ihrer ersten Argumentstelle typisiert ist, verbunden wird.

Die Spezifikation von Properties erfolgt in DAML+OIL nach folgendem Schema:[1)]

```
<   daml:ObjectProperty    |   daml: DatatypeProperty       | daml:TransitiveProperty |
    daml: UniqueProperty   |   daml: UnambiguousProperty>
        <rdfs:domain rdfs:resource="k1"/>
        <rdfs:range    rdfs:resource="k2"/>
</  daml:ObjectProperty    |   daml: DatatypeProperty       | daml:TransitiveProperty |
    daml: UniqueProperty   |   daml: UnambiguousProperty>
```

In DAML+OIL wird zwischen *Objekt-Properties* und *Datentype-Properties* unterschieden.[2)] Beide Property-Arten sind als Subklassen von *rdf:Property* definiert. Objekt-Properties sind solche Properties, die in ihrer zweiten Argumentstelle mit einer DAML+OIL-Klasse typisiert sind. Datentyp-Properties sind hingegen solche Properties, die in ihrer zweiten Argumentstelle mit einem DAML+OIL-Datentyp typisiert sind.

---

1) Das Symbol | in der Spezifikation ist eine Trennlinie zwischen zueinander alternativen Ausdrücken .

2) Vgl. CONNOLLY ET AL. (2001), Abschnitt 2.

Die DAML+OIL-Klasse *daml:ObjectProperty* der Objekt-Properties wird in die Klassen *daml:TransitiveProperty*, *daml:UniqueProperty* und *daml:UnambiguousProperty* unterteilt. Bei Properties, die als Instanzen der DAML+OIL-Klasse *daml: TransitiveProperty* spezifiziert werden, handelt es sich um transitive Relationen.[1)] Mit Instanzen von *daml:UniqueProperty* werden rechtseindeutige Relationen spezifiziert. Somit sind funktionale Relationen in DAML+OIL ausdrückbar. Linkseindeutige Relationen werden hingegen als Instanzen von *daml:UnambiguousProperty* spezifiziert. Darüber hinaus kann die Inversebeziehung zwischen zwei Properties mit Hilfe der Property *daml:inverseOf* spezifiziert werden. Sie ist sowohl in ihrer ersten als auch in ihrer zweiten Argumentstelle mit der DAML+OIL-Klasse *daml:ObjectProperty* spezifiziert.

Schließlich können mit DAML+OIL zwar Integritäts-, aber keine Inferenzregeln konstruiert werden. Für die Konstruktion von Integritätsregeln kann auf die Ausdrucksmittel zurückgegriffen werden, die in der Restriktions-Sektion für Konzeptspezifikationen vorgesehen ist. Darüber hinaus ist es jedoch nicht möglich, Regelbezeichnungen oder -definitionen zu spezifizieren.

5.1.1.7 OWL

Die *Web Ontology Language* (OWL) ist die jüngste unter den vorgestellten Ontologie-Sprachen. Im Januar 2004 ist OWL als W3C-Recommendation verabschiedet worden und wird vermutlich in Zukunft vom W3C zunehmend als Standard für Web-Ontologien propagiert werden.[2)] Dabei wird OWL in den drei „aufwärtskompatiblen“ Ontologie-Sprachen *OWL Lite*, *OWL DL* und *OWL Full* diskutiert. Die Unterscheidung zwischen den drei OWL-Spracharten lässt es offen, die für den jeweiligen Modellierungskontext benötigte Ausdrucksmächtigkeit in Anspruch zu nehmen. Dabei ist OWL Lite gegenüber OWL DL und OWL DL wiederum gegenüber OWL Full in der Ausdrucksmächtigkeit unterlegen. Die gesamte Ausdrucksmächtigkeit von OWL wird erst durch OWL Full erschlossen. Daher wird im Folgenden auch lediglich auf OWL Full eingegangen. Dabei wird weiterhin in einer verkürzten Form lediglich von „OWL“ gesprochen.

Wie bei DAML+OIL auch, werden durch OWL den metasprachlichen Ausdrucksmöglichkeiten von RDF(S) weitere metasprachliche Ausdrucksmöglichkeiten angeschlossen. Somit ist jede RDF(S)-Ontologie auch eine OWL-Ontologie.[3)] Als einzige objektsprachliche Ausdrucksmittel werden hingegen die OWL-Klassen *owl:Thing* und *owl:Nothing* zur Verfügung gestellt, wobei *owl:Thing* und *owl:Nothing* Super- bzw. Subklasse zu allen anderen OWL-Klassen sind.

---

1) Auf Transitivität wurde bereits mehrfach im Vorfeld eingegangen. Wenn eine Relation R transitiv ist, hat zu gelten:

$$\forall x,y,z: (R(x,y) \wedge R(y,z)) \rightarrow R(x,z).$$

2) Vgl. MCGUINESS/VAN HARMELEN (2004).

3) Vgl. MCGUINESS/VAN HARMELEN (2004), Abschnitt 1.3.

Die Spezifikation eines Konzepts k_1 erfolgt in OWL nach folgendem Schema:

```
<owl:Class rdf:ID="k1">
    <owl:equivalentClass> Konzept </daml:equivalentClass>
    <owl:disjointWith> Konzept </owl:disjointWith>
    <owl:intersectionOf> Konzeptliste </owl:intersectionOf>
    <owl:complementOf> Konzept</owl:unionOf>
    <owl:Restriction>
        <owl:onProperty> Property </owl:onProperty>
        Restriktionen
    <owl:/Restriction>
</owl:Class>
```

Konzepte werden als Instanzen der OWL-Klasse *owl:Class* spezifiziert. Sie werden im Folgenden als *OWL-Klassen* bezeichnet. Dabei ist *owl:Class* eine Subklasse von *rdfs:Class*. Daher handelt es sich bei jeder OWL-Klasse auch um eine RDF(S)-Klasse.

Taxonomische Beziehungen zwischen OWL-Klassen können durch *rdfs:SubclassOf* spezifiziert werden. Um die Äquivalenz von OWL-Klassen auszudrücken, wird hingegen die metasprachliche Relation *owl:equivalentClass* verwendet. Um die Inkompatibilität zweier OWL-Klassen auszudrücken, kann *owl:disjointWith* verwendet werden.

Mit *owl:intersectionOf* wird ausgedrückt, dass die Extension eines Konzepts der Schnittmenge der Extensionen von Konzepten aus einer Konzeptliste entspricht. *owl:complementOf* verbindet jeweils ein Konzept mit einem weiteren Konzept. Dabei hat zu gelten, dass aus der Nichtzugehörigkeit eines formalen Objektes zu der Extension des einen Konzepts auf seine Zugehörigkeit zu der Extension des zweiten Konzepts geschlossen werden kann.

In der Sektion im Anschluss an den Ausdruck *owl:Restriction* werden *Restriktionen* ausgedrückt. Hinsichtlich der Spezifikation konzeptspezifischer Restriktionen stimmt die Vorgehensweise mit der in DAML+OIL überein. Auch in OWL werden Restriktionen jeweils in Bezug auf eine Property spezifiziert, deren Angabe nunmehr mit *owl:onProperty* erfolgt. Um propertyspezifische Restriktionen ausdrücken zu können, stehen mehrere metasprachliche Relationen zur Verfügung. Beispielweise kann mit Hilfe von *owl:allValuesFrom* ausgedrückt werden, dass alle Instanzen eines Konzepts nur mit Instanzen aus der Extension des näher bestimmten zweiten Konzepts in der Relation stehen können, auf die die Restriktion bezogen ist. Mit *owl:someValuesFrom* wird hingegen ausgedrückt, dass alle Instanzen des entsprechenden Konzepts mit mindestens einer Instanz aus der Extension des näher bestimmten zweiten Konzepts in der Relation stehen können, auf die die Restriktion bezogen ist.

In der Restriktions-Sektion können darüber hinaus propertyspezifische Kardinalitäten spezifiziert werden. Hierfür stehen die metasprachlichen Relationen *owl:mincardinality*, *owl:maxcardinality* und *owl:cardinality* zur Verfügung. Mit *owl:mincardinality* wird angegeben, mit wie vielen voneinander unterschiedlichen formalen Objekten jede Instanz des Konzepts, für das die Restriktion gültig ist, in der entsprechenden Relation *mindestens* stehen muss. Mit *owl:maxcardinality* wird hingegen angegeben, mit wie vielen voneinander unterschiedlichen formalen Objekten jede Instanz des Konzepts, für das die Restriktion gültig ist, in der entsprechenden Relation *höchstens* stehen kann. Die Angabe zu *owl:cardinality* ist eine Kombination aus übereinstimmender Minimum- und Maximumkardinalität.

Da – wie bei DAML+OIL und RDF(S) auch – OWL-Klassen dem gleichen Individuenbereich wie ihre Instanzen angehören, können in OWL Meta-Konzepte spezifiziert werden. Hierzu wird eine OWL-Klasse in eine *rdf:type*-Beziehung zu einer anderen OWL-Klasse gesetzt. Bei der zweiten OWL-Klasse handelt es sich in diesem Fall um ein Meta-Konzept.

Um mengenwertige Konzepte in OWL ausdrücken zu können, müssen die entsprechenden Ausdrucksmittel aus RDF(S) herangezogen werden. Es handelt sich hierbei um die RDF(S)-Klasse *rdfs:Container* und ihre Subklassen.

Relationen aus einer Ontologie werden in der OWL-Terminologie als *Properties* bezeichnet. Die Spezifikation von Properties erfolgt in OWL nach folgendem Schema:

```
<  owl: DatatypeProperty            | owl:ObjectProperty     | owl:AnnotationProperty   |
   owl:InverseFunctionalProperty | owl:FunctionalProperty | owl:DeprecatedProperty   |
   owl:TransitiveProperty         | owl:SymmetricProperty>
      <rdfs:domain rdfs:resource="k1"/>
      <rdfs:range   rdfs:Resource="k2"/>
</ owl: DatatypeProperty            | owl:ObjectProperty     | owl:AnnotationProperty   |
   owl:InverseFunctionalProperty | owl:FunctionalProperty | owl:DeprecatedProperty   |
   owl:TransitiveProperty         | owl:SymmetricProperty>
```

Eine wesentliche Unterscheidung zwischen Relationen wird in OWL durch die beiden OWL-Klassen *owl:DatatypeProperty* und *owl:ObjectProperty* getroffen. Relationen, die in ihrer zweiten Argumentstelle mit einem Datenkonzept typisiert sind, gehören zur Extension von *owl:DatatypeProperty*. Instanzen von *owl:ObjectProperty* sind hingegen solche Relationen, die in ihrer zweiten Argumentstelle mit einem Domänenkonzept typisiert sind.

owl:AnnotationProperty ist als Klasse aller metasprachlichen Relationen definiert. Somit handelt es sich bei *owl:AnnotationProperty* bei genauerer Diktion um ein *metametasprachliches* Konstrukt. Beispielweise sind die beiden metasprachlichen Relationen *rdfs:Label* und *rdfs:Comment*, die für die metasprachliche Bezeichnung bzw. Definition von Konzepten verwendet werden können, Instanzen von *owl:AnnotationProperty*.[1)]

1) Vgl. BECHHOFER ET AL. (2004), Abschnitt 7.

Bei den Instanzen von *owl:DeprecatedProperty* handelt es sich um Relationen, die in aktuellen Versionen der Ontologie nicht berüksichtigt werden brauchen.

Mit *owl:inverseFunctionalProperty* und *owl:functionalProperty* werden links- bzw. rechtseindeutige Relationen spezifiziert. Wenn eine Relation als Instanz von *owl:inverseFunctionalProperty* angegeben ist, dann dürfen zwei unterschiedliche Instanzen aus der Extension des Argumentkonzepts nicht mit der gleichen Instanz aus der Extension des Zielkonzeptes in entsprechender Relation stehen. Es handelt sich somit bei Instanzen von *owl:inverseFunctionalProperty* um *injektive* Operationen. Umgekehrt darf eine Instanz aus der Extension des Argumentkonzeptes nicht mit zwei unterschiedlichen Instanzen aus der Extension des Zielkonzeptes in der entsprechenden Relation stehen, wenn die Relation eine Instanz von *owl:functionalProperty* ist. Somit handelt es sich bei Instanzen von *owl:functionalProperty* um (rechtseindeutige) Operationen.

Die wichtigsten Ordnungsrelationen können als Instanzen von *owl:TransitiveProperty* oder *owl:SymmetricProperty* spezifiziert werden. Dabei gilt, dass beide OWL-Klassen Subklassen von *owl:ObjectProperty* sind. Somit dürfen nur solche Relationen als Instanzen von *owl:TransitiveProperty* oder *owl:SymmetricProperty* spezifiziert werden, die sowohl in ihrem Argument als auch in ihrem Ziel mit einem Domänenkonzept typisiert sind. Dabei umfasst die Extension von *owl:TransitiveProperty* Relationen, die transitiv sind. Alle Instanzen von *owl:SymmetricProperty* sind symmetrische Relation.

Um eine Ordnung auf der Menge der Relationen ausdrücken zu können, kann in OWL die metasprachliche Relation *rdfs:SubPropertyOf* verwendet werden. Darüber hinaus können Relationen in OWL anhand der beiden metasprachlichen Relationen *owl:equivalentProperty* und *owl:inverseOf* zueinander in Beziehung gesetzt werden. Wenn zwei Relationen in *owl:equivalentProperty*-Beziehung zueinander stehen, müssen die gleichen Paare von Instanzen durch sie verbunden werden. Durch *owl:inverseOf* kann hingegen die Inverse-Beziehung zwischen Relationen ausgedrückt werden.

Für die Bezeichnung und Definition von Relationen können in OWL die metasprachlichen RDF(S)-Relationen *rdfs:Label* und *rdfs:Comment* verwendet werden. Alle OWL-Klassen zur Spezifikation von Relationen sind nämlich Subklassen der RDF(S)-Klasse *rdf:Property* und können somit im Argument der beiden Relationen vorkommen. Jedoch stellt OWL stellt keine Ausdrucksmittel für die Spezifikation regelartiger Zusammenhänge zur Verfügung. Daher können in OWL keine Inferenz- und Integritätsregeln und somit auch keine Regel-Bezeichner und-Definitionen ausgedrückt werden.

5.1.2 Synopsis zu Ontologie-Sprachen

Die Bewertung der Ontologie-Sprachen aus den vorherigen Abschnitten wird in der Tabelle 12 zu einem *synoptischen Überblick* zusammengeführt. Dabei werden die Ausführungen aus den vorherigen Abschnitten den nominal skalierten Anforderungen aus den drei Teilkatalogen zur Beurteilung der statischen Struktur von Ontologie-Sprachen gegenübergestellt.

Mit dem Symbol „✓“ wird angegeben, dass die Ontologie-Sprache über die Ausdrucksmöglichkeit verfügt, die der jeweiligen Anforderung entspricht. Mit dem Symbol „-“wird hingegen angegeben, dass die Ontologie-Sprache über keine Ausdrucksmöglichkeiten verfügt, die der jeweiligen Anforderung entsprechen. Mit ☑ wird schließlich angegeben, dass die Sprache zwar über kein Ausdrucksmittel verfügt, um die entsprechende Anforderung *unmittelbar* zu erfüllen, jedoch die Anforderung durch entsprechende Regeln erfüllt werden kann. Entsprechend setzt die Beurteilung einer Ontologie-Sprache bezüglich einer Anforderung mit ☑ voraus, dass die betroffene Sprache über Ausdrucksmittel zur Konstruktion von Inferenz- oder Integritätsregeln verfügt.

		Ontolingua	F-Logic	SHOE	RDF(S)	DAML+Oil	OWL	$SPEZ_{OS}$
Konzepte	Taxonomie	✓	✓	✓	✓	✓	✓	✓
	Inkompatibilität	☑	☑	☑	-	✓	✓	✓
	Äquivalenz	☑	☑	☑	-	✓	✓	✓
	Meta-Konzepte	✓	✓	-	✓	✓	✓	-
	Konzept-Attribute	✓	✓	-	✓	✓	✓	-
	mengenwertige Konzepte	✓	-	-	✓	✓	✓	✓
	Konzept-Bezeichner	✓	✓	✓	✓	✓	✓	✓
	Konzept-Definition	✓	✓	✓	✓	✓	✓	✓
Relationen	Typisierung von Relationen	✓	✓	✓	✓	✓	✓	✓
	Relationen-Ordnung	✓	☑	-	✓	✓	✓	☑
	n-stellige Relationen	✓	-	-	-	-	-	✓
	Relations-Attribute	✓	✓	-	✓	✓	✓	-
	funktionale Relationen	✓	✓	-	-	✓	✓	✓
	Kardinalitäten	✓	-	-	-	✓	✓	-
	Relationen-Bezeichner	✓	✓	✓	✓	✓	✓	✓
	Relationen-Definition	✓	✓	✓	✓	✓	✓	✓
Regeln	Integritätsregeln	✓	✓	-	-	✓	✓	✓
	Inferenzregeln	✓	✓	✓	-	-	-	✓
	Regel-Bezeichner	-	-	✓	-	-	-	-
	Regel-Definition	-	-	-	-	-	-	✓

Legende		
✓ Anforderung erfüllt	- Anforderung nicht erfüllt	☑ Anforderung durch Inferenzregeln erfüllt

Tabelle 12: Vergleich von Sprachen zur Konstruktion von Ontologien

In der letzten Spalte der Tabelle 12 ist die Beurteilung der Ontologie-Sprache enthalten, die in der vorliegenden Arbeit auf der Basis von ontologischen Spezifikationen entwickelt wurde. Sie wird verkürzt als $SPEZ_{OS}$ angegeben. Wie in der Tabelle zu sehen ist, werden von $SPEZ_{OS}$ alle Anforderungen – außer der Spezifiktion von Meta-Konzepten, von Konzept-Attributen, von Relations-Attributen, von Kardinalitäten und von Regel-Bezeichnern – erfüllt. Im Gegensatz zu den Sprachen, die bezüglich der ersten beiden genannten Anforderungen positiv beurteilt wurden, wird in $SPEZ_{OS}$ eine strikte Separation objekt- und metasprachlicher Ausdrucksmittel beibehalten. Als Instanzen objektsprachlicher Konstrukte kommen hierbei nur formale Konstrukte aus Strukturen in Betracht. Die Instanziierung von objektsprachlichen Konstrukten mit objektsprachlichen Konstrukten ist in $SPEZ_{OS}$ grundsätzlich nicht möglich. Es kommt lediglich die Instanziierung von metasprachlichen Konstrukten mit objektsprachlichen Konstrukten in Betracht. Bezüglich dieses Punktes ist $SPEZ_{OS}$ in dem Sinne erweiterbar, dass ontologische Signaturen um Klassen metasprachlicher Konstrukte erweitert werden könnten. Diese metasprachlichen Konstrukte könnten beispielsweise mit deskriptiven Symbolen instanziiert werden. Eine solche Erweiterung wurde in der vorliegenden Arbeit allerdings nicht berücksichtigt. Denn Ausdrucksmöglichkeiten für Meta-Konzepte und Konzept-Attribute sind mit Klassen metasprachlicher Konstrukte verbunden. Die hiermit verbunden Verkomplizierungen können für die intendierten Anwendungen von $SPEZ_{OS}$ nicht gerechtfertigt werden.

5.2 Evaluation der dynamischen Struktur

Für die Evaluation der dynamischen Struktur des integrativen Modellierungskonzepts wird der Anforderungskatalog aus Abschnitt 2.1.3.2.3 herangezogen. Ontologie-Netze werden im Folgenden anhand der Anforderungen aus dem Anforderungskatalog für die dynamische Struktur evaluiert. Dabei werden für *ordinal* skalierte Anforderungen die Ausprägungen *sehr niedrig*, *niedrig*, *mittelmäßig*, *hoch* und *sehr hoch* verwendet. Für Anforderungen, bezüglich derer nur eine Beurteilung getroffen werden kann, ob sie von Ontologie-Netzen erfüllt werden oder nicht, wird hingegen eine *nominale* Skala mit den Ausprägungen *erfüllt* und *nicht erfüllt* verwendet.[1)]

Wie zuvor angekündigt, verläuft die Evaluation der dynamischen Struktur *konzeptendogen*. Denn in die Evaluation werden nur Alternativen einbezogen, die zu dem Zweck entwickelt wurden, Ontologien so in ein Modellierungskonzept einzubinden, dass kooperative Informationssysteme sowohl hinsichtlich ihrer statischen als auch ihrer dynamischen Aspekte repräsentiert werden können. Allerdings sind bislang keine Konzepte vorgestellt worden, die für eine solche Integration von Ontologien in Frage kämen. Ein konzeptexogener Vergleich von Ontologie-Netzen mit alternativen Modellierungssprachen kommt zudem auch nicht in Frage, da hiermit unweigerlich eine „Verzerrung“ der Evaluation verbunden wäre. Keine der Modellierungssprachen, die dem Verfasser bis-

1) Für eine derartige Evaluation von verschiedenen Petri-Netz-Klasen vgl. ZELEWSKI (1996), S. 372 ff., und die ausführlicheren Erörterungen zu den Kriterienausprägungen in ZELEWSKI (1995), Bd. 9, S. 16 ff.

lang bekannt sind, wurde nämlich zu dem Zweck entwickelt, Ontologien um dynamische Aspekte zu erweitern.

Auch wenn von einer solchen Verzerrung abstrahiert würde,[1)] könnte keine konzeptexogene Evaluation durchgeführt werden, weil sie voraussetzen würde, dass die alternativen Modellierungssprachen mit der gleichen Intensität untersucht worden wären, wie es für Ontologien der Fall ist. Selbst wenn man sich auf die wesentlichen Aspekte der alternativen Modellierungssprachen beschränkt, so würde hierdurch der Rahmen der Arbeit gesprengt werden. Daher werden im Folgenden lediglich Ontologie-Netze evaluiert.

Die Anforderung der Spezifizierbarkeit *einfacher Prozesse* wird von Ontologie-Netzen in sehr hohem Maße erfüllt. Die sequentielle Anordnung von Ereignissen, aufgrund derer eine operationale Variation ontologiegestützter Modelle erfolgt, ist durch Ontologie-Netze ohne Probleme rekonstruierbar. Sie werden durch solche Ontologie-Teilnetze rekonstruiert, die sich bezüglich ihrer Netztopologie auch als allgemeine *Synchronisationsgraphen*[2)] charakterisieren lassen. Bei allgemeinen Synchronisationsgraphen handelt es sich um solche (Teil-)Netze, in denen alle Stellen st_m sowohl in ihren Vorbereichen $VB_{ST}(st_m)$ als auch in ihren Nachbereichen $NB_{ST}(st_m)$ jeweils höchstens[3)] eine Transition aufweisen dürfen. Entsprechend können in Synchronisationsgraphen keine zwei voneinander unterschiedliche Transitionen vorkommen, die bei einem gemeinsamen Schalten von der gleichen Stelle st_m Marken abziehen müssten. Genauso wenig kann es vorkommen, dass sie bei einem gleichzeitigen Schalten auf der gleichen Stelle st_m Marken ablegen müssten. Ontologie-Netze können in ihrer Gesamtheit oder in Teilen als allgemeine Synchronisationsgraphen konstruiert werden. Genauso können sequentielle Prozesse allerdings auch in Ontologie-Netzen stattfinden, in denen Stellen mehrere Vorbereichs- oder Nachbereichstransitionen aufweisen. Entsprechend können einfache Prozesse mit Hilfe von Ontologie-Netzen auf einfache Weise modelliert werden.

Die Spezifizierbarkeit *nebenläufiger Prozesse* gilt als Eigenart von Petri-Netzen, durch die sie von vielen anderen Modellierungskonzepten abgegrenzt werden. Auch für Ontologie-Netze ist die Spezifizierbarkeit von nebenläufigen Prozessen nicht problematisch. Die Nebenläufigkeit wird sogar nicht nur bezüglich der gegenseitigen Abhängigkeit der Aktivierungen von unterschiedlichen Transitionen zugelassen, sondern auch hinsichtlich der unterschiedlichen Grundsubstitutionen, bezüglich derer eine Transition tr_n in einem Ontologie-Netz schalten kann. Somit wird in Ontologie-Netzen nicht nur die *in-*

---

1) Eine konzeptexogene Evaluation findet sich beispielsweise in AOUMEUR (2001), S. 195 ff. Die dort entwickelten CO-NETS werden im Rahmen der Evaluation mit alternativen Modellierungskonzepten verglichen, die allerdings zumeist auf den gleichen Prinzipien wie CO-NETS – nämlich der *Termersetzungs-Logik* – basieren.

2) Vgl. BAUMGARTEN (1996), S. 72.

3) Wird sowohl in dem Vorbereich $VB_{ST}(st_m)$ als auch in dem Nachbereich $NB_{ST}(st_m)$ jeweils nur *genau* eine Transition zugelassen, handelt es sich um einen *speziellen* Synchronisationsgraphen; vgl. BAUMGARTEN (1996), S. 72.

tertransitionale, sondern auch die *intratransitionale* Nebenläufigkeit unterstützt. Lediglich bezüglich des nebenläufigen Zugriffs auf mengenwertige Terme weisen Ontologie-Netze Schwächen auf. Denn Ontologie-Netze erlauben nicht den nebenläufigen Zugriff von Transitionen auf Terme, aus denen sich die mengenwertigen Terme in Marken zusammensetzen können. Es können stets nur die Marken in ihrer Gesamtheit abgezogen und abgelegt werden. Jedoch können Bedingungen formuliert werden, denen einwertige Individuen genügen müssen, die durch Terme aus einem mengenwertigen Term aus einer Marke repräsentiert werden. Der Zugriff auf die einwertigen Terme erfolgt beispielsweise durch das Relationssymbol *Element_of*, anhand dessen überprüft werden kann, ob das entsprechende einwertige Individuum Element des mengenwertigen Individuums ist. Ebenso können Operationssymbole angewandt werden, die durch Operationen extensional interpretiert werden, die für den Zugriff auf Mengen geeignet sind. In Abbildung 37 ist beispielsweise ein Ontologie-Netz graphisch dargestellt, in dem das Operationssymbol *Intersection* verwendet wird, um die Schnittmenge SET_3 von zwei Mengen SET_1 und SET_2 zu bestimmen.

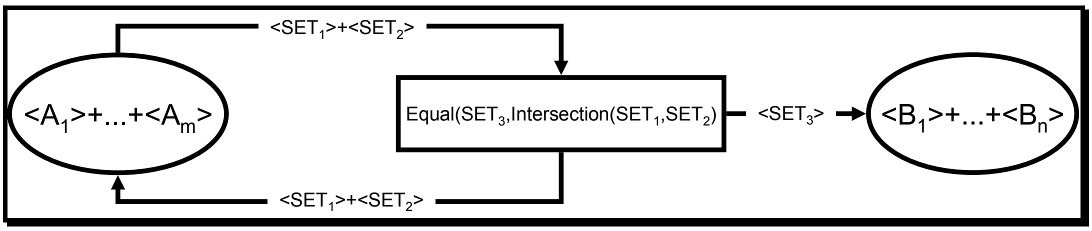

Abbildung 37: Zugriff auf mengenwertige Terme in Ontologie-Netzen

Genau so, wie Ontologie-Netze die Modellierung komplexer Prozesse erlauben, erlauben sie auch die Modellierung von *Entscheidungsalternativen*, um in bestimmten Zuständen die Durchführbarkeit verschiedener Ereignisse abzubilden. Entscheidungsalternativen spiegeln sich in Ontologie-Netzen zumeist in konfliktionär zueinander aktivierten Transitionen wider. In Ontologie-Netzen können die Stellen durchaus so markiert sein, dass verschiedene Transitionen alternativ zueinander schalten können. Die Konsequenzen, die sich aus der Entscheidung zu Gunsten einer beliebigen Alternative ergeben, entsprechen den Folgemarkierungen und den darin aktivierten Transitionen.

Die Möglichkeit, voneinander getrennte Teilprozesse zu synchronisieren, ist eine ausgewiesene Stärke aller Petri-Netz-Klassen. Zur Handhabung solcher komplexer Prozesse erlauben auch Ontologie-Netze die Synchronisation von Teilprozessen in sehr hohem Maße. Beispielsweise lassen sich Teilprozesse synchronisieren, indem zwei Teilprozesse durch eine Transition tr_n berandet werden. Die Vorbereichsstellen der Transition tr_n erstrecken sich in diesem Fall u.a. auf die beiden zu synchronisierenden Teilprozesse. Erst wenn beide Teilprozesse zu einer derartigen Markierung der entsprechenden Stellen führen, kann tr_n schalten, da alle Vorbereichsstellen von tr_n den Markenkonsum durch tr_n sicherstellen müssen. Darüber hinaus erstreckt sich das Synchronisationspotenzial von Ontologie-Netzen auch auf den Austausch von Marken zwischen Teilprozessen. Transitionen aus unterschiedlichen Teilprozessen können in Ontologie-Netzen – wie in anderen Petri-Netzen auch – gegenseitig auf die Stelle zugreifen.

Kontrollvariablen, von denen die Durchführung von Prozessen abhängig gemacht werden soll und die je nach Prozessablauf variiert werden, können in Ontologie-Netzen mühelos berücksichtigt werden. Dabei können die zu berücksichtigenden Variablen in Ontologie-Netzen sowohl qualitativer als auch quantitativer Ausprägungen sein. Sowohl im Fall der qualitativen als auch im Fall der quantitativen Kontrollvariablen wird die Gültigkeit derjenigen Formel, die aus der Grundsubstitution der Kontrollvariablen in der Transitionsannotation hervorgeht, in dem Support $A_{SIG_{ST}}$ überprüft. Beispielsweise kann im Fall eines iterativen Prozesses entsprechend Abbildung 38 der Inhalt einer Marke <X> auf der Stelle st_1 ausgewertet werden. Aufgrund der gemeinsam kontradiktorischen Transitionsannotationen $AN_{TR}(tr_1)$ bzw. $AN_{TR}(tr_2)$ können die Transitionen tr_1 und tr_2 niemals gleichzeitig aktiviert werden.

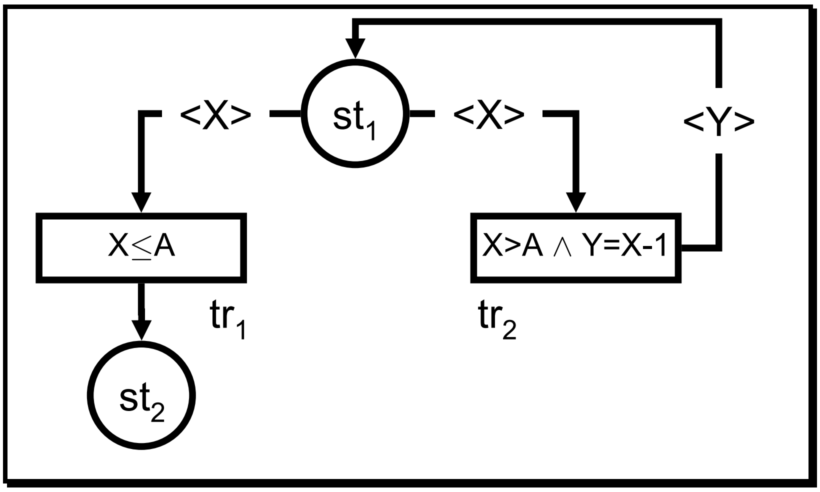

Abbildung 38: Ontologie-Netz für iterative Prozesse

Wenn die Bedingung für tr_1 erfüllt ist, terminiert der Prozess durch eine Markierung der Stelle st_2. Wenn hingegen die Bedingung für tr_2 erfüllt ist, wird die Stelle st_1 erneut mit einer neuen Markenkopie markiert und entweder tr_1 oder tr_2 erneut aktiviert.

Sowohl die Schaltvoraussetzungen als auch die Schaltwirkungen von Transitionen sind in Ontologie-Netzen präzise festgelegt. Bezüglich der Schaltvoraussetzungen ist in Ontologie-Netzen im Vergleich zu sonstigen Petri-Netz-Klassen sogar eine wesentliche Besonderheit auszumachen. Sie betrifft die explizite Unterscheidung der verschiedenen Transitionsarten bei der Überprüfung von Aktivierungsbedingungen. Zu diesem Zweck werden in Ontologie-Netzen unterschiedlichen Transitionsarten auch unterschiedliche Aktivierungsprioritäten eingeräumt. Durch die Aktivierungsprioritäten für Transitionsarten werden die *Preconditions* – also die Bedingungen, die vor dem Eintreten von Ereignissen erfüllt sein müssen – auf differenzierte Art berücksichtigt. Zudem ist das das potenzielle Eintreten von Ereignissen durch die Trennung der statischen Schaltvoraussetzungen von den dynamischen Schaltvoraussetzungen kontrollierbar.

Über Preconditions hinaus werden auch *Postconditions* – also die Bedingungen, die *nach* dem Eintreten von Ereignissen gültig zu sein haben – in Ontologie-Netzen durch die Ex-ante-Bestimmbarkeit der Schaltwirkungen von Transitionen definiert. Durch die Schaltfunktion SF_{ON} ist eindeutig bestimmt, welche Folgemarkierung M_f einzutreten hat, wenn eine Transition tr_n unter einer Grundsubstitution θ und einer Referenzmarkierung M_r schaltet.

Durch die Aktivierungsprioritäten wird das Kriterium der *Priorisierung deklarativer Regeln* durch Ontologie-Netze erfüllt. Zur Berücksichtigung von deklarativen Regeln wurden für Ontologie-Netze zwei algorithmische Verfahren vorgestellt,[1)] anhand derer Inferenz- und Integritätsregeln, in denen dynamische Relationssymbole vorkommen, in Inferenz- bzw. Integritätstransitionen transformiert werden können.[2)] Dadurch geht zwar der deklarative Charakter der Regelverarbeitung verloren, diesem Umstand wird jedoch durch eine differenzierte Behandlung von deklarativen und prozeduralen Transition in Form von Aktivierungsprioritäten begegnet. Demnach können keine prozeduralen Transitionen aktiviert werden und entsprechend auch nicht schalten, wenn unter der betrachteten Markierung mindestens eine deklarative Transition aktiviert ist.

Bezüglich der *konzeptendogenen Objektgenerierung* weisen Ontologie-Netze eine konzeptionelle Schwäche auf. Ontologie-Netze setzen nämlich einen Support $A_{SIG_{ST}}$ voraus, in dem die Objektmenge OB im Vorfeld bestimmt ist. Außerhalb dieser Objektmenge OB können konzeptendogen keine Individuen erzeugt werden. Somit sind auch Ontologie-Netze nicht in der Lage, Ereignisse zu berücksichtigen, anhand derer formale Objekte erzeugt würden. Diese konzeptionelle Schwäche betrifft allerdings nur die semantischen Aspekte von Ontologie-Netzen. Ontologie-Netze sind jedoch durchgehend mit syntaktischen Konstrukten annotiert. Oftmals wird eine syntaktische Objektgenerierung bereits als ausreichend empfunden, um höheren Petri-Netzen die Fähigkeit der konzeptendogenen Objektgenerierung zu attestieren.[3)] In solchen Fällen reicht es in Ontologie-Netzen aus, Stellen zu spezifizieren, die mit Konstantensymbolen markiert sind, die als Namen der bereits vorhandenen Individuen dienen. Bei Bedarf für eine neue Markenkopie wird durch eine Transition dieser Stelle eine Markenkopie entzogen. In diesem Zusammenhang erweist sich die Möglichkeit der Filterung von Markenkopien durch entsprechende Kantenannotationen als äußerst vorteilhaft. Der Stelle, in der die Terme vorgehalten werden, können nämlich durch gefilterte Kantenannotationen Markenkopien nach Bedarf entzogen werden. Wenn beispielsweise eine Stelle st_m mit Markenkopien des Typs $w_1 \in K^+$ markiert ist, kann eine Transition tr_n über eine Kante (st_m, tr_n) mit einer solchen Kantenannotation, die einen Typ $w_2 \in K^+$ mit $w_2 \preceq w_1$ hat, gefiltert Markenkopien von st_m abziehen.

---

1) Vgl. Abschnitt 4.2.2.1.2.

2) Inferenz- und Integritätsregeln, in denen nur statische Relationssymbole vorkommen, werden bereits dadurch priorisiert, dass in einem Ontologie-Netz ON nur ein Support $A_{SIG_{OS}}$ verwendet werden kann, in dem alle „statischen" Inferenz- und Integritätsregeln gültig sind.

3) Vgl. AOUMEUR (2001), S. 52 f.; AOUMEUR/SAAKE (2002), S. 153 ff.

Wie in S/T-Netzen auch, können in Ontologie-Netzen Kapazitäten für Stellen angegeben werden, die durch stellenbezogene Markierungen nicht überschritten werden dürfen. Insofern bieten Ontologie-Netze grundsätzlich die Möglichkeit, Kapazitäten zu spezifizieren. Jedoch ist diese Kapazitätsrestriktion nur bedingt zufrieden stellend. Sie unterscheidet sich nämlich nicht von der Kapazitätsrestriktion, die für anonyme Marken in S/T-Netzen gültig ist. Im Fall individueller Marken, wie sie von Ontologie-Netzen verwendet werden, kann es auch wünschenswert sein, Kapazitäten für Markenarten zu formulieren, so dass beispielsweise nur eine bestimmte Anzahl von Dokumenten einer bestimmten Art auf der Stelle abgelegt werden kann, auf der grundsätzlich Dokumente jeder Art abgelegt werden können. Solche Restriktionen können zwar durch sehr umständliche Transitionsannotationen „simuliert" werden, allerdings bleibt die Gesamtlösung hierdurch nicht zufrieden stellend.

Eine weitere Schwäche von Ontologie-Netzen offenbart sich bei der Berücksichtigung *zeitbezogener Determinanten*. Ontologie-Netze sind grundsätzlich nicht in der Lage, zeitbezogene Determinanten zu berücksichtigen. Diese Schwäche resultiert aus dem konzeptionellen Defizit der Petri-Netz-gestützten Modellierung. Das Petri-Netz-Konzept basiert nämlich auf dem Prinzip der kausalen Abhängigkeit von Ereignissen. Temporale Abhängigkeiten werden in der Grundausrichtung nicht berücksichtigt.[1)]

Grundsätzlich ist die Implementierbarkeit von Ontologie-Netzen als hoch einzustufen. Denn Ontologie-Netze sind durchgehend in *formaler* Weise ausgedrückt. Der formale *Rahmen* erstreckt sich hierbei auf die konventionelle und sortierte Prädikatenlogik sowie auf Multimengen. Jedoch werden in Ontologie-Netzen Konstrukte verwendet, die weder in der konventionellen noch in der sortierten Prädikatenlogik berücksichtigt werden. Insofern kann auf bereits vorhandene informationstechnische Umsetzungen höherer Petri-Netze nur so weit zurückgegriffen werden, wie es der jeweils verwendete prädikatenlogische Dialekt erlaubt, metasprachliche Ausdrucksmittel zu spezifizieren. Dem Verfasser sind allerdings bislang keine Software-Pakete bekannt, in denen neben den metasprachlichen Ausdrucksmitteln, die bereits „verdrahtet" in der Software verwendet werden, zusätzliche Ausdrucksmittel spezifiziert werden können. Beispielsweise kom-

---

1) Vgl. ZELEWSKI (1995), Bd. 6, S. 120. Ausnahmen hierzu stellen *Timed Petri-Nets* dar; vgl. ADAM ET AL. (1998), S. 143 ff.; BOWDEN (2000), S. 55 ff.; GIRAULT/VALK (2003), S. 115 f.; TACKEN (2001), S. 54 ff.; ZELEWSKI (1995), Bd. 6, S. 119 ff. Dabei kann die Berücksichtigung temporaler Aspekte sich nahezu auf alle Netz-Komponenten erstrecken. In der Regel werden Marken mit einem Zeitstempel versehen, von denen die Schaltprioritäten für Transitionen abhängig gemacht werden. Zudem wird in Timed Petri-Nets oftmals das Prinzip der atomaren Durchführung von Ereignissen durchbrochen, indem Transitionen eine *Schaltdauer* eingeräumt wird. Ebenso können oftmals für Stellen „Verweildauern" und für Kanten „Flussdauern" spezifiziert werden. Im Fall von Verweildauern sind die Marken auf einer Stelle so lange „unsichtbar", bis eine stellenspezifische Dauer abgelaufen ist. Im Fall von Flussdauern wird der Fluss einer Marke über eine Flusskante verzögert. Im Rahmen von Timed Petri Nets existieren unter der Bezeichnung „Hybrid Petri Nets" auch Petri-Netz-Klassen, in denen sowohl diskrete als auch stetige Zeitstrukturen berücksichtigt werden können; vgl. BALDUZZI ET AL. (2001), S. 258 ff.; DAVID/ALLA (2001), S. 11 ff.

men zwar Software-Pakete aus dem Umfeld der *Order Sorted Logic* in Betracht, um taxonomische Beziehungen zwischen Konzepten zu spezifizieren.[1)] Arbeiten aus diesem Umfeld sind bisweilen mehrmals als Grundlage für die Implementierung höherer Petri-Netze vorgestellt worden.[2)] Allerdings erlaubt es keines der hierauf aufbauenden Instrumente, die weiteren metasprachlichen Strukturierungsrelationen auszudrücken, mit denen die Ausdrucksvielfalt von Ontologien erschlossen wird.

Eine wesentliche Stärke von Ontologie-Netzen liegt darin, eine graphische Visualisierbarkeit der erstellten Informationsmodelle zu gewährleisten. Ontologie-Netze können mittels visueller Graphen auf äußerst benutzerfreundliche Weise repräsentiert werden. Dabei werden trotz der graphischen Repräsentation die Vorzüge aus der formalen Präzision von Ontologie-Netzen bewahrt. Denn die graphischen Symbole stehen in isomorphem Verhältnis zu den Netzkomponenten, die sie repräsentieren. Dabei wird die Benutzerfreundlichkeit der graphischen Repräsentation durch Ontologie-Netze in der Hinsicht ausgeschöpft, als dass sie für die unterschiedlichen Stellen- und Transitionsarten auch unterschiedliche Arten von graphischen Symbolen zur Verfügung stellen.

Darüber hinaus wird mit der graphischen Repräsentierbarkeit von Ontologie-Netzen eine Kompatibilität mit der statischen Struktur bewahrt. Denn auch für Ontologien kann die graphische Repräsentation durch semantische Netze oder durch hyperbolische Graphen im Fall der computergestützten Illustration in Anspruch genommen werden. Die Kompatibilität wird dadurch gewahrt, dass beide Repräsentationsarten auf visuelle Graphen zurückgreifen, wodurch potenzielle Benutzer von Ontologie-Netzen keinem „Paradigmenwechsel“ ausgesetzt werden müssen.

1) Vgl. HAMEL/GOGUEN (1994), S. 132 ff.

2) Vgl. BIBERSTEIN ET AL. (2001), S. 74 ff.; STEHR ET AL. (2001), S. 271 ff. Vgl. darüber hinaus WEBER (2002), S. 27 ff., für einen Überblick über computergestützte Instrumente zu verschiedenen Petri-Netzen-Klassen.

Kriterien	nominale Skala						
		ordinale Skala					
	nicht erfüllt	sehr niedrig	niedrig	mittel-mäßig	hoch	sehr hoch	erfüllt
einfache Prozesse						❍	
komplexe Prozesse					❍		
Entscheidungs-alternativen						❍	
Synchronisation						❍	
iterative Prozesse						❍	
Pre- und Postconditions							❍
Priorisierung deklarativer Regeln							❍
konzeptendogene Objektgenerierung				❍			
Kapazitäten					❍		
zeitbezogene Determinanten	❍						
graphische Visualisierbarkeit						❍	
Implementierbarkeit					❍		

Tabelle 13: Evaluation der dynamischen Struktur

6 Fazit

Als erstes intendiertes Ergebnis der vorliegenden Arbeit wurde in Abschnitt 1.1 die Spezifikation einer *Ontologie-Sprache* definiert. Zu diesem Zweck wurden in Abschnitt 3.1 sowohl ein formales als auch ein informales Verständnis von Ontologien vorgestellt.

Bezüglich des informalen Verständnisses von Ontologien wurde hervorgehoben, dass Ontologien formalsprachliche Spezifikationen sprachlicher Ausdrucksmittel sind. Natürlichsprachliche Konstrukte wurden somit als Ontologien ausgeschlossen. Entsprechend wurde hervorgehoben, dass es für die Konstruktion von Ontologien einer formalen Ontologie-Sprache bedarf. Eine derartige formale Ontologie-Sprache wurde innerhalb der Darlegung des formalen Verständnisses von Ontologien in den Abschnitten 3.1.2, 3.1.3 und 3.1.4 vorgestellt.

Bezüglich des formalen Verständnisses wurde – basierend auf den Prinzipien der konventionellen und der sortierten Prädikatenlogik – das Konzept *ontologischer Signaturen* vorgestellt. In ontologischen Signaturen werden die sprachlichen Ausdrucksmittel zur Verfügung gestellt, mit denen *ontologische Ausdrücke* konstruiert werden können. Bei ontologischen Ausdrücken handelt es sich um Zeichenketten, die mit den Zeichen aus dem ontologischen Alphabet $ALPH_{OS}$ konstruiert werden. Damit eine Zeichenkette über $ALPH_{OS}$ als ontologischer Ausdruck zulässig ist, müssen die Operations- und Relationssymbole entsprechend ihrer Typisierung auf ontologische Terme angewendet werden.

Als wesentliche Erweiterung von Ontologien gegenüber der konventionellen und der sortierten Prädikatenlogik wurden die metasprachlichen Ausdrucksmittel zur Berücksichtigung der intensionalen Semantik von objektsprachlichen Ausdrucksmitteln herausgearbeitet. Bei den metasprachlichen Ausdrucksmitteln handelt sich in erster Linie um die Strukturierungsrelationen $\sqsubseteq$, $\circeq$ und $\curlyvee$. Mit Hilfe der Subkonzeptrelation $\sqsubseteq$ können Konzepte aus Ontologien in taxonomische Beziehungen zueinander gesetzt werden. Entsprechend der taxonomischen Struktur auf der Menge K aller Konzepte, können die konzeptspezifischen Termmengen in Inklusionsbeziehungen zueinander stehen. Die Gleichheit oder Disjunktheit von konzeptspezifischen Termmengen wird hingegen mit Hilfe der Äquivalenzrelation $\circeq$ bzw. der Inkompatibilitätsrelation $\curlyvee$ ausgedrückt.

Über die Strukturierungsrelation hinaus sind in ontologischen Signaturen weitere metasprachliche Ausdrucksmittel enthalten, um die intensionale Semantik objektsprachlicher Ausdrucksmittel zu bestimmen. Es handelt sich hierbei um die sprachspezifischen Bezeichnungs- und Definitionsfunktionen. Im Gegensatz zu den Strukturierungsrelationen wird mit Hilfe der sprachspezifischen Bezeichnungs- und Definitionsfunktionen die intensionale Semantik von sprachlichen Ausdrucksmitteln *informal* bestimmt.

Zusätzlich zu den *syntaktischen* und *semantischen* Aspekten von Ontologien wurden in Abschnitt 3.1.4 ihre *pragmatischen* Aspekte herausgearbeitet. Bezüglich dieser verwendungsorientierten Sichtweise wurde insbesondere auf *Inferenz-* und *Integritätsregeln* eingegangen. Es wurde in diesem Kontext herausgearbeitet, dass die Unterscheidung zwischen Inferenz- und Integritätsregeln weder auf der syntaktischen noch auf der semantischen Ebene möglich ist. Inferenz- und Integritätsregeln unterscheiden sich voneinander durch die unterschiedlichen Zwecksetzungen, die mit ihrer Spezifikation verknüpft ist. Während Inferenzregeln dazu verwendet werden, in einer Faktenbasis enthaltene implizite Fakten zu explizieren, werden Integritätsregeln für die Zulässigkeitsprüfung von Fakten verwendet.

Im Rahmen der pragmatischen Aspekte von Ontologien wurde zudem auf Erweiterungsmöglichkeiten für Ontologien eingegangen. Es handelt sich hierbei um solche Erweiterungen, die auch innerhalb des integrativen Modellierungskonzepts in Anspruch genommen werden. Zum einen wurde auf die *Substitution* ontologischer Ausdrücke eingegangen. Sie wird durchgeführt, indem Variablen, die in einem ontologischen Term vorkommen, durch ontologische Terme erstetzt werden. Zum anderen wurden *ontologische Termtupel* als weitere Ausdrucksvariante neben ontologischen Termen und ontologischen Formeln vorgestellt. Über die Substitution und ontologische Termtupel hinaus wurden in Abschnitt 3.1.4.3.3 Möglichkeiten der Differenzierung von Relationssymbolen vorgestellt. Einerseits wurde hierbei aufgezeigt, wie Relationssymbole eingeführt werden können, die aus der Negation von Relationssymbolen hervorgehen. Andererseits wurde aufgezeigt, wie Relationssymbole danach unterschieden werden können, ob sie eine zustandsvariante oder -invariante Extension haben.

Als zweites intendiertes Ergebnis wurde in Abschnitt 1.1 die Entwicklung eines *integrativen Modellierungskonzepts* formuliert. Zu diesem Zweck wurde in Abschnitt 4 das formale Gerüst für *Ontologie-Netze* entwickelt. Bei Ontologie-Netzen handelt es sich um eine Klasse höherer Petri-Netze, deren Komponenten mit Ausdrücken über einer Ontologie annotiert werden. Dadurch leisten Ontologie-Netze die Integration vo einerseits *Ontologien*, die sich zur Spezifikation der *sprachlichen Ausdrucksmittel* für die Modellierung kooperativer Informationssysteme eignen, und andererseits von *Petri-Netzen*, die auf die Modellierung insbesondere der *dynamischen* Aspekte von Geschäftsprozessen zugeschnitten sind.

Aufgrund der Verwandtschaft von Ontologien mit prädikatenlogischen und algebraischen Spezifikationen weisen Ontologie-Netze eine Nähe zu Pr/T-Netzen und algebraischen Netzen auf. Mit Pr/T-Netzen haben Ontologie-Netze gemeinsam, dass sie über das gesamte Ausdrucksvermögen der Prädikatenlogik erster Ordnung verfügen. Bezüglich ihrer integritätsbewahrenden Sorten- bzw. Konzeptstruktur ähneln Ontologie-Netze zudem algebraischen Netzen.

Im Rahmen der Entwicklung von Ontologie-Netzen war es notwendig, Verfahren zur Berücksichtigung der Regelkomponente aus einer Ontologie zu entwickeln. Zu diesem Zweck wurden Verfahren vorgestellt, die die Transformation von Inferenz- und Integritätsregeln aus einer Ontologie in Ontologie-Teilnetze erlauben, mit denen Regeln operational angewendet werden können. Hierdurch wird die Funktionalität von deklarativen Regeln in allen Zuständen des ontologiegestützten Modells bewahrt.

Das dritte intendierte Ergebnis bestand in einer *Evaluation* des integrativen Modellierungskonzepts. Zu diesem Zweck wurde das entwickelte Modellierungskonzept sowohl hinsichtlich seiner *statischen* als auch hinsichtlich seiner *dynamischen* Struktur evaluiert.

Bezüglich der Evaluation der statischen Struktur wurde die Ontologie-Sprache, die für die Spezifikation von Ontologien vorausgesetzt wurde, mit den Ausdrucksmitteln von aktuell diskutierten Ontologie-Sprachen verglichen. Im Rahmen dieses Vergleichs konnte nachgewiesen werden, dass keine der Sprachen eine Dominanz gegenüber dem hier verwendeten Ansatz ausübt.

Im zweiten Schritt wurde das Modellierungskonzept bezüglich seiner dynamischen Struktur beurteilt. Hierbei haben sich Schwächen des Modellierungskonzepts bezüglich dreier Anforderungen ergeben. Aus den Schwächen von Ontologien ergeben sich weiterhin *offene Forschungsfragen*, die im Rahmen der vorliegenden Arbeit nicht weiter thematisiert werden konnten. Erstens wurde aufgezeigt, dass die Forderung nach konzeptendogener Objektgenerierung von Ontologie-Netzen nicht hinreichend unterstützt wird. Um „neue“ Objekte im Modell berücksichtigen zu können, müssen in Ontologie-Netzen Stellen vorgesehen werden, in denen entsprechende Marken vorgehalten werden. Als zweites Defizit von Ontologie-Netzen wurde die fehlende Berücksichtigung zeitbezogener Determinanten ausgemacht. Schließlich wurde aufgezeigt, dass sich Ontologie-Netze zwar grundsätzlich implementieren lassen, jedoch aktuell keine Instrumente vorliegen, mit deren Hilfe Ontologie-Netze konstruiert werden könnten.

Anhang

Anhang A: Referenzontologie und deren Rekonstruktion

Referenzontologie

K:	Person,Unternehmen,Entitaet;	
OPS:	Arbeitet_fuer:	Person → Unternehmen;
RS:	Tochterunternehmen_von:	Unternehmen Unternehmen
	Mutterunternehmen_von:	Unternehmen Unternehmen
	Verbunden_mit:	Unternehmen Unternehmen;
$VAR_{Unternehmen}$:	{x,y};	
bez_{ger}:	Person	→ {Person}
	Unternehmen	→ {Unternehmen}
	Entitaet	→ {Entität}
	Arbeitet_fuer	→ {Arbeitet für}
	Tochterunternehmen_von	→ {Tochterunternehmen von}
	Mutterunternehmen_von	→ {mutterunternehmen von}
	Verbunden_mit	→ {verbunden mit};
$INF_{SIG_{OS}}$:	F_1 :⇔ ∀x,y: mutterunternehmen_von(x,y) ↔ tochterunternehmen_von(y,x),	
	F_2 :⇔ ∀x,y,z: tochterunternehmen_von(x,z) ∧ tochterunternehmen_von(y,z) → verbunden_mit (x,y).	

def_{ger}:	Person	→ Menschlicher Akteur,
	Unternehmen	→ Ökonomische Institution mit dem Ziel der Fremdbedarfsdeckung
	Arbeitet_fuer	→ Arbeitsverhältnis zwischen einer Person und höchstens einem Unternehmen
	Tochterunternehmen_von	→ konzernrechtliche Unterordnung eines Unternehmens gegenüber einem anderen Unternehmen
	Mutterunternehmen_von	→ konzernrechtliche Überordnung eines Unternehmens gegenüber einem anderen Unternehmen
	Verbunden_mit	→ konzernrechtliche Verbundenheit von Unternehmen
	F_1	→ Ein Unternehmen ist genau dann ein Mutterunternehmen eines zweiten Unternehmens, wenn das zweite Unternehmen ein Tochterunternehmen des ersten Unternehmens ist
	F_2	→ Wenn zwei Unternehmen Tochterunternehmen eines dritten Unternehmens sind, dann sind die beiden Unternehmen miteinander verbunden.

In der Referenz-Ontologie werden objektsprachliche Ausdrucksmittel formalsprachlich erfasst, die für die Beschreibung von Beziehungen zwischen einerseits Personen und Unternehmen und andererseits zwischen Unternehmen untereinander von Bedeutung sind. Zum einen kann mit der Ontologie die Arbeits-Beziehung zwischen einer Person und einem Unternehmen ausgedrückt werden.[1)] Zum anderen können konzernrechtliche Beziehungen zwischen Unternehmen ausgedrückt werden. Es handelt sich hierbei zunächst um die Mutter- oder. Tochterunternehmen-Beziehung, die zwischen Unternehmen eines Konzerns vorliegen kann. Darüber hinaus können formale Objekte aus der Extension des Konzepts *Unternehmen* zueinander in einer *verbunden_mit*-Beziehung stehen.

In der Referenz-Ontologie sind neben deskriptiven Symbolen noch zwei Inferenzregeln als objektsprachliche Konstrukte aufgeführt. Mit der ersten Inferenzregel wird der inverse Zusammenhang zwischen den Relationssymbolen *mutterunternehmen_von* und

1) Zwecks Einfachheit wurde hierbei unterstellt, dass jede Person nur für ein Unternehmen arbeiten kann. Daher wurde die *arbeitet_fuer*-Beziehung als Operationssymbol mit einem einwertigen Zielkonzept spezifiziert.

tochterunternehmen_von ausgedrückt. Mit der zweiten Regel wird schließlich ausgedrückt, dass zwei Unternehmen dann miteinander verbunden sind, wenn sie beide Tochterunternehmen desselben Unternehmens sind.

Ontolingua

```
(define-ontology  company)

(define-class       entitaet (?X))

(define-class       person (?X)
 „Konzept menschlicher Akteur“
     :axioms        (Subclass-of   person entitaet))

(define-class       unternehmen (?X)
“Konzept ökonomische Institution mit dem Ziel der Fremdbedarfsdeckung“
     :axioms        (Subclass-of   person entitaet))

(define-function  arbeitet_fuer (?X)
“Funktionale Relation, um Arbeitsverhältnisse zwischen Personen und Unternehmen auszudrücken“
     :axioms        (domain    arbeitet_fuer person)
                    (range     arbeitet_fuer unternehmen))

(define_relation   tochterunternehmen_von (?X ?Y)
“Relation, um die konzernrechtliche Unterordnung eines Unternehmens gegenüber
einem anderen Unternehmen auszudrücken“
     :axioms        (domain    tochterunternehmen_von unternehmen)
                    (range     tochterunternehmen_von unternehmen))

(define_relation   mutterunternehmen_von (?X ?Y)
“Relation, um die konzernrechtliche Überordnung eines Unternehmens gegenüber
einem anderen Unternehmen auszudrücken“
     :axioms        (domain    mutterunternehmen_von unternehmen)
                    (range     mutterunternehmen_von unternehmen))
                    (forall ?X ?Y
                    (<=>    (mutterunternehmen_von ?X ?Y)
                            (tochterunternehmen_von ?Y ?X)))

(define_relation   verbunden_mit (?X ?Y)
“Relation, um die konzernrechtliche Verbundenheit von Unternehmen auszudrücken“
     :axioms        (domain    verbunden_mit unternehmen)
                    (range     verbunden_mit unternehmen)
                    (forall ?X ?Y ?Z
                    (=>     (verbunden_mit ?X ?Y)
                            (tochterunternehmen_von ?X ?Z) AND
                            (tochterunternehmen_von ?Y ?Z))).
```

F-Logic

```
Person          ::Entitaet.
Unternehmen     ::Entitaet.

Person        [ arbeitet_fuer=>Unternehmen].
Unternehmen [ tochterunternehmen_von=>>Unternehmen;
                verbunden_mit=>>Unternehmen;
                mutterunternehmen_von=>>Unternehmen].

FORALL X,Y:   X[mutterunternehmen_von->>Y]
              ↔ Y[tochterunternehmen_von->>X].
FORALL X,Y,Z: X[tochterunternehmen_von->>Z] AND Y[tochterunternehmen_von->>Z]
              → X[verbunden_mit->>Y].

Person        [ hat_Bezeichnung->>{„Person“};
                hat_Definition->„menschlicher Akteur“].

Unternehmen [ hat_Bezeichnung->>{„Unternehmen“};
                hat_Definition->>„ökonomische Institution mit dem Ziel der Fremdbedarfsdeckung“].
arbeitet_fuer [ hat_Bezeichnung->>{„arbeitet_für“};
                hat_Definition->„ Relation, um Arbeitsverhältnisse zwischen Personen und
                Unternehmen auszudrücken“].

tochterunternehmen_von[
                hat_Bezeichnung->>{„Tochterunternehmen von“};
                hat_Definition->„Relation, um die konzernrechtliche Unterordnung eines
                Unternehmens gegenüber einem anderen Unternehmen auszudrücken“].

mutterunternehmen_von[
                hat_Bezeichnung->>{„Mutterunternehmen von“};
                hat_Definition->„Relation, um die konzernrechtliche Überordnung eines
                Unternehmens gegenüber einem anderen Unternehmen auszudrücken“].

verbunden_mit[  hat_Bezeichnung->>{„verbunden mit“};
                hat_Definition->„Relation, um die konzernrechtliche Verbundenheit von
                Unternehmen auszudrücken“].
```

SHOE

```
<ONTOLOGY ID=„Unternehmen“>

<DEF-CATEGORY NAME      =„Entitaet“
        [SHORT          =„Entitaet“>
</DEF-CATEGORY>

<DEF-CATEGORY NAME=„Person“
        [ISA            =„Entitaet“]
        [DESCRIPTION    =„Menschliche Akteure“]
        [SHORT          =„Person“]>
</DEF-CATEGORY>

<DEF-CATEGORY NAME=„Unternehmen“
        [ISA            =„Entitaet“]
        [DESCRIPTION    =„ökonomische Institution mit dem Ziel der Fremdbedarfsdeckung“]
        [SHORT          =„Unternehmen“]>
</DEF-CATEGORY>

<DEF-RELATION NAME =„arbeitet_fuer“
        [DESCRIPTION    =  „Relation, um Arbeitsverhältnisse zwischen Personen und Unternehmen
                           auszudrücken“]
        [SHORT          =„arbeitet für“]>
    <DEF-ARG        POS=„FROM“   TYPE=„Person“>
    <DEF-ARG        POS=„TO“     TYPE=„Unternehmen“>
</DEF-RELATION>

<DEF-RELATION NAME=„tochterunternehmen_von“
        [DESCRIPTION =  „Relation, um die konzernrechtliche Unterordnung eines Unternehmens
                        gegenüber einem anderen Unternehmen auszudrücken“]
        [SHORT       =„tochterunternehmen von“]>
    <DEF-ARG        POS=„FROM“   TYPE=„Unternehmen“>
    <DEF-ARG        POS=„TO“     TYPE=„Unternehmen“>
</DEF-RELATION>
```

```
<DEF-RELATION NAME=„mutterunternehmen_von“
        [DESCRIPTION =   „Relation, um die konzernrechtliche Überordnung eines Unternehmens
                          gegenüber einem anderen Unternehmen auszudrücken“]
        [SHORT          =„mutterunternehmen von“]>
    <DEF-ARG            POS=„FROM“   TYPE=„Unternehmen“>
    <DEF-ARG            POS=„TO“     TYPE=„Unternehmen“>
</DEF-RELATION>

<DEF-RELATION NAME=„verbunden_mit“
        [DESCRIPTION =   „Relation, um die konzernrechtliche Verbundenheit von Unternehmen
                          auszudrücken]
        [SHORT          =„verbunden mit“]>
    <DEF-ARG            POS=„FROM“   TYPE=„Unternehmen“>
    <DEF-ARG            POS=„TO“     TYPE=„Unternehmen“>
</DEF-RELATION>

<DEF-INFERENCE
        [DESCRIPTION=    „inverse Beziehung zwischen „mutterunternehmen_von“
                          und „tochterunternehmen von“ 1. “]>
    <INF-IF>
        <RELATION NAME=„tochterunternehmen_von“>
            <ARG POS=„1“ VALUE=„x“>
            <ARG POS=„2“ VALUE=„y“>
        </RELATION>
    </INF-IF>
    <INF-THEN>
        <RELATION NAME=„mutterunternehmen_von“>
            <ARG POS=„1“ VALUE=„y“>
            <ARG POS=„2“ VALUE=„x“>
        </RELATION>
    </INF-THEN>
</DEF-INFERENCE>
```

```
<DEF-INFERENCE
        [DESCRIPTION=    „inverse Beziehung zwischen „mutterunternehmen_von“
                         und „tochterunternehmen_von“ 2. “]>
    <INF-IF>
        <RELATION NAME=„mutterunternehmen_von“>
            <ARG POS=„1“ VALUE=„x“>
            <ARG POS=„2“ VALUE=„y“>
        </RELATION>
    </INF-IF>
    <INF-THEN>
        <RELATION NAME=„tochterunternehmen_von“>
            <ARG POS=„1“ VALUE=„y“>
            <ARG POS=„2“ VALUE=„x“>
        </RELATION>
    </INF-THEN>
</DEF-INFERENCE>

<DEF-INFERENCE
        [DESCRIPTION=    „Wenn zwei Unternehmen Tochterunternehmen eines dritten Unternehmens
                         sind, dann sind die beiden Unternehmen miteinander verbunden“]>
    <INF-IF>
        <RELATION NAME=„tochterunternehmen_von“>
            <ARG POS=„1“ VALUE=„x“>
            <ARG POS=„2“ VALUE=„z“>
        </RELATION>
        <RELATION NAME=„tochterunternehmen_von“>
            <ARG POS=„1“ VALUE=„y“>
            <ARG POS=„2“ VALUE=„z“>
        </RELATION>
    </INF-IF>
    <INF-THEN>
        <RELATION NAME=„verbunden_mit“>
            <ARG POS=„1“ VALUE=„x“>
            <ARG POS=„2“ VALUE=„y“>
        </RELATION>
    </INF-THEN>
</DEF-INFERENCE>
</ONTOLOGY>
```

RDF(S)

```
<rdf:RDF
    xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
    xmlns:rdfs='http://www.w3.org/2000/01/rdf-schema#'>

<rdf:Description rdf:ID='Entitaet'>
    <rdf:type rdfs:Resource='http://www.w3.org/2000/01/rdf-schema#Class'/>
    <rdfs:Label xml:lang='ger'>Entitaet</rdfs:Label>
</rdf:Description>

<rdf:Description rdf:ID='verbunden_mit'>
<rdfs:Comment>
Relation, um die konzernrechtliche Verbundenheit von Unternehmen auszudrücken
</rdfs:Comment>
<rdf:type rdfs:Resource='http://www.w3.org/1999/02/22-rdf-syntax-ns#Property'/>
    <rdfs:Label xml:lang='ger'>verbunden mit</rdfs:Label>
    <rdfs:domain rdfs:Resource='Unternehmen'/>
    <rdfs:range rdfs:Resource='Unternehmen'/>
</rdf:Description>

<rdf:Description rdf:ID='mutterunternehmen_von'>
<rdfs:Comment>
Relation, um die konzernrechtliche Überordnung eines Unternehmens gegenüber einem anderen
Unternehmen auszudrücken
</rdfs:Comment>
<rdf:type rdfs:Resource='http://www.w3.org/1999/02/22-rdf-syntax-ns#Property'/>
    <rdfs:Label xml:lang='ger'>mutterunternehmen von</rdfs:Label>
    <rdfs:domain rdfs:Resource='Unternehmen'/>
    <rdfs:range rdfs:Resource='Unternehmen'/>
</rdf:Description>

<rdf:Description rdf:ID='Unternehmen'>
<rdfs:Comment>
ökonomische Institution mit dem Ziel der Fremdbedarfsdeckung
</rdfs:Comment>
<rdf:type rdfs:Resource='http://www.w3.org/2000/01/rdf-schema#Class'/>
    <rdfs:Label xml:lang='ger'>Unternehmen</rdfs:Label>
    <rdfs:SubClassOf rdfs:Resource='Entitaet'/>
</rdf:Description>
```

```
<rdf:Description rdf:ID='Person'>
<rdfs:Comment>
menschlicher Akteur
</rdfs:Comment>
<rdf:type rdfs:Resource='http://www.w3.org/2000/01/rdf-schema#Class'/>
    <rdfs:Label xml:lang='ger'>Person</rdfs:Label>
    <rdfs:SubClassOf rdfs:Resource='Entitaet'/>
</rdf:Description>

<rdf:Description rdf:ID='arbeitet_fuer'>
<rdfs:Comment>
Relation, um Arbeitsverhältnisse zwischen Personen und Unternehmen auszudrücken
</rdfs:Comment>
<rdf:type rdfs:Resource='http://www.w3.org/1999/02/22-rdf-syntax-ns#Property'/>
    <rdfs:Label xml:lang='ger'>arbeitet_fuer</rdfs:Label>
    <rdfs:domain rdfs:Resource='Person'/>
    <rdfs:range rdfs:Resource='Unternehmen'/>
</rdf:Description>

<rdf:Description rdf:ID='tochterunternehmen_von'>
<rdfs:Comment>
Relation, um die konzernrechtliche Unterordnung eines Unternehmens gegenüber einem anderen
Unternehmen auszudrücken
</rdfs:Comment>
<rdf:type rdfs:Resource='http://www.w3.org/1999/02/22-rdf-syntax-ns#Property'/>
    <rdfs:Label xml:lang='ger'>tochterunternehmen_von</rdfs:Label>
    <rdfs:domain rdfs:Resource='Unternehmen'/>
    <rdfs:range rdfs:Resource='Unternehmen'/>
</rdf:Description>
</rdf:RDF>
```

DAML+OIL

```
<daml:Class rdf:ID="#Entitaet"/>

<daml:Class rdf:ID="#Person">
    <rdfs:Label xml:lang="ger">#Person</rdfs:Label>
    <rdfs:SubClassOf rdfs:Resource="#Entitaet"/>
    <daml:SubClassOf>
        <daml:Restriction daml:minCardinality="0" daml:maxCardinality="1">
            <daml:onProperty rdfs:Resource="#arbeitet_fuer"/>
            <daml:toClass rdfs:Resource="#Unternehmen"/>
        </daml:Restriction>
    </daml:SubClassOf>
</daml:Class>
```

```
<daml:Class rdf:ID="#Unternehmen">
    <rdfs:Comment>
    ökonomische Institution mit dem Ziel der Fremdbedarfsdeckung
    </rdfs:Comment>
    <rdfs:Label xml:lang="ger">#Unternehmen</rdfs:Label>
    <rdfs:SubClassOf rdfs:Resource="#Entitaet"/>
    <daml:SubClassOf>
        <daml:Restriction daml:minCardinality="0">
            <daml:onProperty rdfs:Resource="#tochterunternehmen_von"/>
            <daml:toClass rdfs:Resource="#Unternehmen"/>
        </daml:Restriction>
    </daml:SubClassOf>
    <daml:SubClassOf>
        <daml:Restriction daml:minCardinality="0">
            <daml:onProperty rdfs:Resource="#verbunden_mit"/>
            <daml:toClass rdfs:Resource="#Unternehmen"/>
        </daml:Restriction>
    </daml:SubClassOf>
    <daml:SubClassOf>
        <daml:Restriction daml:minCardinality="0">
            <daml:onProperty rdfs:Resource="#mutterunternehmen_von"/>
            <daml:toClass rdfs:Resource="#Unternehmen"/>
        </daml:Restriction>
    </daml:SubClassOf>
</daml:Class>

<daml:ObjectProperty rdf:ID="#arbeitet_fuer">
    <rdfs:Comment>
    Relation, um Arbeitsverhältnisse zwischen Personen und Unternehmen auszudrücken
    </rdfs:Comment>
    <rdfs:Label xml:lang="ger">#arbeitet_fuer</rdfs:Label>
    <daml:inverseOf daml:ObjectProperty="#hat_Mitarbeiter">
</daml:ObjectProperty>
```

```
<daml:TransitiveProperty rdf:ID="#mutterunternehmen_von">
    <rdfs:Comment>
        Relation, um die konzernrechtliche Überordnung eines Unternehmen gegenüber einem anderen
        Unternehmen auszudrücken
    </rdfs:Comment>
    <rdfs:Label xml:lang="ger">#mutterunternehmen_von</rdfs:Label>
    <daml:inverseOf daml:ObjectProperty="#tochterunternehmen_von">
</daml:TransitiveProperty>

<daml:TransitiveProperty rdf:ID=tochterunternehmen_von>
    <rdfs:Label xml:lang="ger">"tochterunternehmen von"</rdfs:Label>
    <rdfs:Comment>
        Relation, um die konzernrechtliche Unterordnung eines Unternehmens gegenüber einem anderen
        Unternehmen auszudrücken
    </rdfs:Comment>
    <daml:inverseOf daml:ObjectProperty="#mutterunternehmen_von">
</daml:ObjectProperty>

<daml:ObjectProperty rdf:ID=verbunden_mit>
    <rdfs:Comment>
        Relation, um die konzernrechtliche Verbundenheit von Unternehmen auszudrücken
    </rdfs:Comment>
    <rdfs:Label xml:lang="ger">"verbunden mit"</rdfs:Label>
</daml:ObjectProperty>
```

OWL

```
<owl:Class rdf:ID="Entitaet"/>

<owl:Class rdf:ID="#Person">
    <rdfs:Label xml:lang="ger">#Person</rdfs:Label>
    <rdfs:SubClassOf rdfs:Resource="#Entitaet"/>
    <owl:SubClassOf>
        <owl:Restriction owl:minCardinality="0" owl:maxCardinality="1">
            <owl:onProperty rdfs:Resource="#arbeitet_fuer"/>
            <owl:toClass rdfs:Resource="#Unternehmen"/>
        </owl:Restriction>
    </owl:SubClassOf>
</owl:Class>
```

```
<owl:Class rdf:ID="#Unternehmen">
    <rdfs:Comment>
    ökonomische Institution mit dem Ziel der Fremdbedarfsdeckung
    </rdfs:Comment>
    <rdfs:Label xml:lang="ger">#Unternehmen</rdfs:Label>
    <rdfs:SubClassOf rdfs:Resource="#Entitaet"/>
    <owl:SubClassOf>
        <owl:Restriction daml:minCardinality="0">
            <owl:onProperty rdfs:Resource="#tochterunternehmen_von"/>
            <owl:toClass rdfs:Resource="#Unternehmen"/>
        </owl:Restriction>
    </owl:SubClassOf>
    <owl:SubClassOf>
        <owl:Restriction owl:minCardinality="0">
            <owl:onProperty rdfs:Resource="#verbunden_mit"/>
            <owl:toClass rdfs:Resource="#Unternehmen"/>
        </owl:Restriction>
    </owl:SubClassOf>
    <owl:SubClassOf>
        <owl:Restriction owl:minCardinality="0">
            <owl:onProperty rdfs:Resource="#mutterunternehmen_von"/>
            <owl:toClass rdfs:Resource="#Unternehmen"/>
        </owl:Restriction>
    </owl:SubClassOf>
</owl:Class>

<owl:ObjectProperty rdf:ID="#hat_Mitarbeiter">
    <rdfs:Label xml:lang="ger">#arbeitet_fuer</rdfs:Label>
    <owl:inverseOf owl:ObjectProperty="#arbeitet_fuer">
</owl:ObjectProperty>

<owl:TransitiveProperty rdf:ID=mutterunternehmen_von>
    <rdfs:Comment>
        Relation, um die konzernrechtliche Überordnung eines Unternehmen gegenüber einem anderen
        Unternehmen auszudrücken
    </rdfs:Comment>
    <rdfs:Label xml:lang="ger">"mutterunternehmen von"</rdfs:Label>
</owl:TransitiveProperty>
```

```
<owl:TransitiveProperty rdf:ID=tochterunternehmen_von>
    <rdfs:Label xml:lang="ger">"tochterunternehmen von"</rdfs:Label>
    <rdfs:Comment>
        Relation, um die konzernrechtliche Unterordnung eines Unternehmens gegenüber einem anderen
        Unternehmen auszudrücken
    </rdfs:Comment>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID=verbunden_mit>
    <rdfs:Comment>
        Relation, um die konzernrechtliche Verbundenheit von Unternehmen auszudrücken
    </rdfs:Comment>
    <rdfs:Label xml:lang="ger">"verbunden mit"</rdfs:Label>
</owl:ObjectProperty>
```

Anhang B: Fallstudie

In der Fallstudie werden Anwendungsfälle (Use Cases) repräsentiert, die typisch für Informationssysteme in Banken sind. Hierzu gehören: *Anlegen von Kunden*, *Eröffnen von Konten*, *Vornehmen von Überweisungen* und *Erhöhung von Zinssätzen*. Sämtliche der Anwendungsfälle werden durch prozedurale Transitionen modelliert. Darüber hinaus existieren in dem Ontologie-Netz (vgl. Abbildung 39) zwei deklarative Transitionen. Mit einer Integritätstransition, die durch Transformation aus einer Inferenzregel aus der zugrunde gelegten Ontologie konstruiert wurde, wird überprüft, ob die Angaben zu einerseits Konten und andererseits zu Kunden zulässig sind. Sie sind dann unzulässig, wenn mindestens in der Stelle, durch die Wissen über Konten repräsentiert wird, ein Konto einem Kunden zugeordnet ist, in dessen Kontenmenge das entsprechende Konto nicht vorkommt. Darüber hinaus wird durch eine Inferenztransition eine solche Geschäftsregel umgesetzt, derzufolge allen Kunden ein Kreditangebot unterbreitet wird, wenn sie 80% ihres Dispositionskredits ausgeschöpft haben.

In der zu Grunde gelegten Ontologie wird zwischen *Giro*- und *Kreditkonten* als Subkonzepten des Konzepts *Konto* unterschieden. Sämtliches Wissen über Konten wird gemeinsam auf einer Stelle (*Konten*) vorgehalten. Durch filternde Kantenannotation wird gewährleistet, dass für Anwendungsfälle keine Konten verwendet werden dürfen, für deren Art der Anwenungsfall unzulässig ist. Beispielsweise wird durch die Flusskante, die die Stelle *Konten* mit der Transition *Überweisung* verbindet, gewährleistet, dass keine Überweisungen von Girokonten zu Kreditkonten oder zu Konten im Allgemeinen erfolgen. Dies liegt daran, dass die Variabe GTO aus der Kantenannotation nur mit Termen zum Konzept *Girokonto* substituiert werden kann. Die Variable KTO kann hingegen mit Termen zum Konzept *Konto* und somit auch mit allen Termen zu den Konzepten *Girokonto* und *Kreditkonto* substituiert werden. Darüber hinaus ist es Kunden aus dem Fallbeispiel erlaubt, mehrere Konten beliebiger Art gleichzeitig oder auch keines zu führen. Dies wird dadurch umgesetzt, dass die letzte Argumentstelle des Relationssysmbols *Kundendaten* mit dem mengenwertigen Term MEN(Konto) typisiert ist.

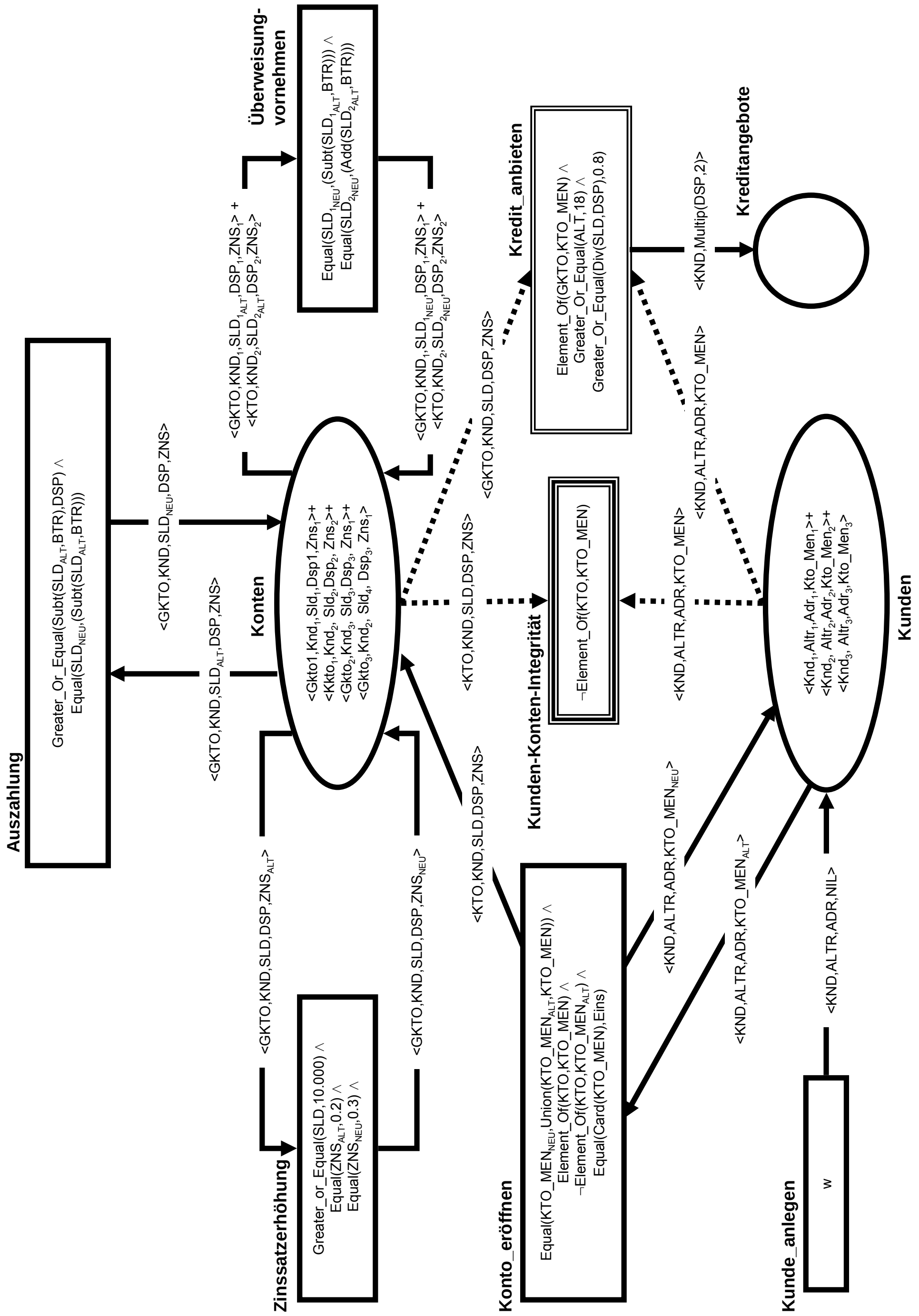

Abbildung 39: Visueller Graph zum Ontologie-Netz aus der Fallstudie

$SPEZ_{OS}$

SIG_{OS}

// Einwertigen Konzepte //

K_{EW}: $\top, \bot$,
Konto, Girokonto, Kreditkonto,
Kunde, Alter,
Saldo, Betrag, Dispo, Zinssatz,
Schranke_EW,
Schranke_MW,
Schranke_EW_Domaenenkonzept,
Schranke_MW_Domaenenkonzept,
Schranke_EW_Datenkonzept,
Schranke_MW_Datenkonzept,
Real, $Real_{pos}$, $Real_{neg}$,
Integer, $Integer_{POS}$;

// Mengenwertiger Konzepte //

K_{MW}: MEN(Konto);

// Subkonzeptrelation //

$\sqsubseteq$: Girokonto $\sqsubseteq$ Konto,
Kreditkonto $\sqsubseteq$ Konto,
Saldo $\sqsubseteq$ Real
Betrag $\sqsubseteq$ Real
Dispo $\sqsubseteq$ $Real_{neg}$,
Zinssatz $\sqsubseteq$ $Real_{pos}$
Adresse $\sqsubseteq$ String
$Integer_{POS}$ $\sqsubseteq$ Integer
Alter $\sqsubseteq$ $Integer_{POS}$
$Real_{neg}$ $\sqsubseteq$ Real
($Real_{pos}$ $\sqsubseteq$ Real
$Integer_{pos}$ $\sqsubseteq$ $Real_{pos}$
Schranke_EW_Domaenenkonzept $\sqsubseteq$ Schranke_EW
Schranke_EW_Datenkonzepte $\sqsubseteq$ Schranke_EW
Schranke_MW_Domaenenkonzept $\sqsubseteq$ Schranke_MW
Schranke_MW_Datenkonzept $\sqsubseteq$ Schranke_MW

// Inkompatibilitätsrelation //

$\curlyvee$: Girokonto $\curlyvee$ Kreditkonto;

// Operationssymbole //

OPS: Multip: Real Real $\rightarrow$ Real
Div: Real $Real_{POS}$ $\rightarrow$ Real
Subt: Real Real $\rightarrow$ Real
Union: Schranke_MW Schranke_MW $\rightarrow$ Schranke_MW
Card: Schranke_MW $\rightarrow$ Integer;

// Kostantensymbole – Werden benötigt, um die Grundttermtupel zur Annotation von Stellen im Ontologie-Netz zu konstruieren //

$KONF_{SIG_{OS}}$

$KON_{Kreditkonto} = \{ Gkto_1, Gkto_2, ..., Gkto_{|KON_{Girokonto}|} \}$

$KON_{Kreditkonto} = \{ Kkto_1, Kkto_2, ..., Kkto_{|KON_{Kreditkonto}|} \}$

$KON_{Kunde} = \{ Knd_1, Knd_2, ..., Knd_{|KON_{Kunde}|} \}$

$KON_{Adresse} = \{ Adr_1, Adr_2, ..., Adr_{|KON_{Adresse}|} \}$

$KON_{MEN(Konto)} = \{ Kto\_Men_1, Kto\_Men_2, ..., Kto\_Men_{|KON_{MEN(Konto)}|}, NIL \}$

$KON_{Saldo} = \{ Sld_1, Sld_2, ..., Sld_{[KON_{Saldo}|} \}$

$KON_{Dispo} = \{ Dsp_1, Dsp_2, ..., Dsp_{|KON_{Dispo}|} \}$

$KON_{Zinssatz} = \{ Zns_1, Zns_2, ..., Zns_{|KON_{Zinssatz}|} \}$

$KON_{Alter} = \{ Altr_1, Altr_2, ..., Altr_{|KON_{Alter}|} \}$

$KON_{Real} = \{ 0.00, ..., 1000000.00 \}$;

// Reelle Zahlen werden in der Signatur SIG_{OS} und in der statischen SIG_{OS}-Struktur $A_{SIG_{ST}}$ gleich bezeichnet//

RS

RS_{ST}: Equal: $\top\ \top \rightarrow \top$
Element_of: Schranke_EW Schranke_MW;

RS_{DY}: Kundendaten: Kunde Alter Adresse MEN(Konto)
Kontendaten: Konto Kunde Saldo Dispo Zinssatz
Kreditangebotsdaten: Kunde Betrag;

// Variablen – Werden benötigt, um die Informations- und Flusskanten aus dem Ontologie-Netz zu annotieren //

$VARF_{SIG_{OS}}$

VAR_{Konto} $=\{$ KTO$\}$,

$VAR_{Girokonto}$ $=\{$ GKTO$\}$,

VAR_{Saldo} $=\{$ $SLD, SLD1_{NEU}, SLD1_{ALT}, SLD2_{NEU}, SLD2_{ALT}, SLD_{NEU}, SLD_{ALT}$ $\}$,

VAR_{Kunde} $=\{$ KND, KND_1, KND_2 $\}$,

VAR_{Dispo} $=\{$ DSP, DSP_1, DSP_2 $\}$,

$VAR_{Zinssatz}$ $=\{$ $ZNS, ZNS_{ALT}, ZNS_{NEU}, ZNS_1, ZNS_2$ $\}$,

VAR_{Betrag} $=\{$ BTR$\}$,

$VAR_{MEN(Konto)}$ $=\{$ KTO_MEN, KTO_MEN_{ALT}, KTO_MEN_{NEU} $\}$;

VAR_{Alter} $=\{$ ALTR$\}$;

// Inferenzregeln //

$INF_{SIG_{OS}}$

INF: ∀ KND, ALT, ADR, KTO_MEN, KTO, SLD, DSP, ZNS:
Kundendaten(KND,ALT,ADR,KTO_MEN) ∧
Element_Of(GKTO,KTO_MEN)
Kontendaten(GKTO,KND,SLD,DSP,ZNS) ∧
Greater_Or_Equal(ALT,18) ∧
Greater_Or_Equal(Div(SLD,DSP),0.8)
→ Kreditangebotsdaten(KND,Multip(DSP,2));

// Integritätsregeln //

$INT_{SIG_{OS}}$

INT: ∀ KND, ALT, ADR, KTO_MEN, KTO, SLD, DSP, ZNS:
Kundendaten(KND,ALT,ADR,KTO_MEN) ∧
Kontendaten(KTO,KND,SLD,DSP,ZNS)
→ Element_of(KTO,KTO_MEN);

// deutsche Bezeichnungen//

bezf

bez_{ger}:	Konto	→	{Konto, Bankkonto, Kontobuch}
	Girokonto	→	{Girokonto}
	Kreditkonto	→	{Kreditkonto}
	Kunde	→	{Kunde, Klient, Mandant}
	Alter	→	{Alter}
	Saldo	→	{Saldo}
	Dispo	→	{Dispo}
	Kundendaten	→	{Kundendaten}
	Kontendaten	→	{Kontendaten}
	Kreditangebot	→	{Kreditangebot};

// englische Bezeichnungen //

bez_{eng} ={	Konto	→	{account}
	Girokonto	→	{current-account}
	Kreditkonto	→	{charge-account}
	Kunde	→	{customer,client}
	Alter	→	{age}
	Saldo	→	{balance}
	Dispo	→	{overdraft-facility}
	Kundendaten	→	{customer data}
	Kontendaten	→	{account data}
	Kreditangebot	→	{charge data};

// deutsche Definitionen //

def_{ger}:	Konto	→	kundenbezogener Datensatz zu Guthaben und kurz- oder langfristigen Verbindlichkeiten
	Girokonto	→	kundenbezogener Datensatz zu Guthaben oder kurzfristigen Verbindlichkeiten
	Kreditkonto	→	kundenbezogener Datensatz zu langfristigen Verbindlichkeiten
	INF	→	Kunden, die mindestens 80% ihres Dispos ausgeschöpft haben, muss ein Kreditangebot vorgelegt werden
	INT	→	Konten die zwar einem Kunden zugeordnet sind, aber nicht in deren Kontenmenge vorkommen, sind unzulässig);

// englische Definitionen //

def_{eng}: Konto → `customer-specific data record on assets and short- or long-term courtesies`

Girokonto → `customer-specific data record on assets and short-term courtesies`

Kreditkonto → `customer-specific data record on assets and long-term courtesies`

INF → `customers, who exhaust at least 80% of their overdraft-facilty,have to be offered a credit`

INT → `account assigned to customers without incidencing in the account-set of the customer are inadmissible;`

// Ontologie-Netz //

ON

// Stellen //

ST : Konten, Kunden, Kreditangebote;

// Stellenannotation //

AN_{ST}: Konten → Kontendaten

Kunden → Kundendaten

Kreditangebote → Kreditangebotsdaten;

// Transitionen //

TR

// deklarative Transitionen //

TR_{DEKL}

TR_{INT} Konten-Kunden-Integrität

TR_{INF} Kredit_anbieten;

// prozedurale Transitionen //

TR_{PROZ} Überweisung, Zinssatzerhöhung, Kunde_anlegen, Konto_eröffnen, Auszahlung;

// Transitionszuordnungsfunktionen //

$trans_{INF}$ INF → Kredit_anbieten;

$trans_{INT}$ INT → Konten-Kunden-Integrität;

// Transitionsannotation //

AN_{TR}: Konten-Kunden-Integrität → ¬Element_Of(KTO,KTO_MEN)

Kredit_anbieten → Element_Of(KTO,KTO_MEN) ∧
¬Greater_or_Equal(ALT,18) ∧
Equal(Div(SLD,DSP),0.8))

Überweisung → $Equal(SLD_{1_{NEU}},(Subt(SLD_{1_{ALT}},SUM))) \wedge$
$Equal(SLD_{2_{NEU}},(ADD(SLD_{2_{ALT}},SUM))))$

Zinssatzerhöhung → Greater_Or_Equal(SLD,10.000) ∧
$Equal(ZNS_{ALT},0.2) \wedge EQUAL(ZNS_{NEU},0.3))$

Kunde_anlegen → w

Konto_eröffnen → $Equal\ (KTO\_MEN_{NEU},$
$Union(KTO\_MEN_{ALT},KTO\_MEN)) \wedge$
Element_Of(KTO,KTO_MEN) ∧
$\neg Element\_Of(KTO,KTO\_MEN_{ALT}) \wedge$
Equal(Card(KTO_MEN),1)

Auszahlung → $Greater\_or\_Equal(Subt(SLD_{ALT},BTR),DSP) \wedge$
$Equal(SLD_{NEU},(SUBT(SLD_{ALT},BTR)))$

// Flussrelation //

FR: (Kredit_anbieten,Kreditangebote)
(Konten,Überweisung)
(Überweisung,Konten)
(Konten,Zinssatzerhöhung)
(Zinssatzerhöhung,Konten)
(Konto_eröffnen,Konten)
(Kunden,Konto_eröffnen)
(Konto_eröffnen,Kunden)
(Kunde_anlegen,Kunden);

// allgemeine Flusskantenannotation //

AN_{FR}: ((Kredit_anbieten,Kreditangebote),
<KND,Multip(DSP,2)>),

((Konten,Überweisung),
$<GKTO,KND_1,SLD_{1_{ALT}},DSP_1,ZNS_1> + <KTO,KND_2,SLD_{2_{ALT}},DSP_2,ZNS_2>),$

((Überweisung,Konten),
$<GKTO,KND_1,SLD_{1_{NEU}},DSP_1,ZNS_1> + <KTO,KND_2,SLD_{2_{NEU}},DSP_2,ZNS_2>),$

((Konten,Zinssatzerhöhung),
$<GKTO,KND,SLD,DSP,ZNS_{ALT}>),$

((Zinssatzerhöhung,Konten),

<GKTO,KND,SLD,DSP,ZNS_{NEU}>),

((Konto_eröffnen,Konten),

<KTO,KND,SLD,DSP,ZNS>),

((Kunden,Konto_eröffnen),

<KND,ALT,ADR,KTO_MEN_{ALT}>),

((Konto_eröffnen,Kunden),

<KND,ALT,ADR,KTO_MEN_{NEU}>),

((Kunde_anlegen,Kunden)}

<KND,ALT,ADR,NIL>)};

// Informationsrelation //

IR: (Konten,Konten-Kunden_Integrität),

(Kunden,Konten-Kunden_Integrität),

(Konten,Kredit_anbieten),

(Kunden,Kredit_anbieten)};

// Informationskantenannotation //

AN_{IR} ={ (Kunden, Konten-Kunden_Integrität),

<KND,ALT,ADR,KTO_MEN>),

(Konten, Konten-Kunden_Integrität),

<KTO,KND,SLD,DSP,ZNS>),

(Konten, Kredit_anbieten),

<GKTO,KND,SLD,DSP,ZNS>),

(Kunden, Kredit_anbieten)

<KND,ALT,ADR,KTO_MEN>)};

// Support //

$A_{SIG_{ST}}$

OBF_{OS}

OB_{Kunde}	={ a_mueller, w_meier, s_schulze,.....}
$OB_{GiroKonto}$	={ 0075417328, 0037458136, 0023321246,.....}
$OB_{Kreditkonto}$	={ 0083645173,....}
OB_{Konto}	$= OB_{Girokonto} \cup OB_{Kreditkonto}$
$OB_{Integer}$	$= \mathbb{Z}$
$OB_{Integer_{pos}}$	$= \mathbb{Z}_{+}$
OB_{Real}	$= \mathbb{R}$
$OB_{Real_{neg}}$	$= \mathbb{R}_{-}$
$OB_{Real_{pos}}$	$= \mathbb{R}_{+}$
OB_{Dispo}	= [-10000,....,0]
$OB_{Zinssatz}$	= [0.0,.....,0.5]
OB_{Alter}	= [0,....,120]
OB_{Saldo}	= [-200.000,.....,10000000]
$OB_{Adresse}$	$= OB_{String}$
$OB_{MEN(Konto)}$	$= pot(OB_{Konto})$

OPF

multip : $OB_{REAL} \times OB_{REAL} \rightarrow OB_{REAL}$,
$multip(ob_1,ob_2)=ob_1*ob_2$;

div : $OB_{REAL} \times OB_{REAL_{POS}} \rightarrow OB_{REAL}$,
$div(ob_1,ob_2)=ob_1/ob_2$;

add : $OB_{REAL} \times OB_{REAL} \rightarrow OB_{REAL}$,
$multip(ob_1,ob_2)=ob_1+ob_2$;

subs : $OB_{REAL} \times OB_{REAL} \rightarrow OB_{REAL}$,
$multip(ob_1,ob_2)=ob_1-ob_2$;

union : $OB_{Schranke\_MW} \times OB_{Schranke\_MW} \rightarrow OB_{Schranke\_MW}$,
$union(ob_1,ob_2)=ob_1 \cup ob_2$;

card : $OB_{Schranke\_MW} \rightarrow OB_{Integer_{pos}}$
$card(ob) = |ob|$.

RF_{ST}

equal: = $OB_T \times OB_T \rightarrow OB_T$,
$equal=\{(ob_1,ob_2) \mid (ob_1=ob_2)$;

element_of: = $OB_{Schranke\_EW} \times OB_{Schranke\_MW}$,

element_of$\{(ob_1,ob_2) \mid ob_1 \in ob_2\}$.

// Extensionale Interpretation der Konstantensymbole aus SIG_{OS} //

I_{OPS}

Eins $\rightarrow$1

Knd_1 $\rightarrow$ a_mueller

Knd_2 $\rightarrow$ w_meier

Knd_3 $\rightarrow$ s_schulze

....

$Gkto_1$ $\rightarrow$ 0075417328

$Gkto_2$ $\rightarrow$ 0037458136

$Gkto_3$ $\rightarrow$ 0023321246

$Kkto_1$ $\rightarrow$ 0083645173

....

Adr_1 $\rightarrow$ „Schadowstr. 41, 45141 Essen“

Adr_2 $\rightarrow$ „Obergath 5, 45147 Essen“

Adr_3 $\rightarrow$ „Langenbeckstr. 73, 45141 Essen“

....

$Kto\_Men_1 \rightarrow \{0075417328\}$

$Kto\_Men_2 \rightarrow \{0023321246,0083645173\}$

$Kto\_Men_3 \rightarrow \{0037458136\}$

....

$Sld_1 \rightarrow -900$

$Sld_2 \rightarrow 0$

$Sld_3 \rightarrow 12000$

$Sld_3 \rightarrow 200$

....

$Dsp_1 \rightarrow -1000$

$Dsp_2 \rightarrow -22000$

$Dsp_3 \rightarrow -2000$

$Dsp_3 \rightarrow -1000$

....

$Zns_1 \rightarrow 0.2$

$Zns_2 \rightarrow 0.4$

....

$Altr_1 \rightarrow 21$

$Altr_2 \rightarrow 32$

$Altr_3 \rightarrow 45;$

// Stellenbezogene Anfangsmarkierungen – Werden mit Grundtermtupeln konstruiert. Für die Konstruktion der Grundtermtupel werden die Konstantentsymbole aus SIG_{OS} verwendet //

M_0

Konten $\rightarrow <Gkto_1,Knd_1,Sld_1,Dsp_1,Zns_1>$
$+<Kkto_1,Knd_2,Sld_2,Dsp_2,Zns_2>$
$+<Gkto2,Knd_3,Sld_3,Dsp_3,Zns_1>$
$+<Gkto3,Knd_2,Sld_4,Dsp_4,Zns_1>;$

Kunden $\rightarrow <Knd_1,Altr_1,Adr_1,Kto\_Men_1>$
$+<Knd_2,Altr_2,Adr_2,Kto\_Men_2>$
$+<Knd_3,Altr_3,Adr_3,Kto\_Men_3>;$

Kreditangebote $\rightarrow \lambda.$

// Auswertungen der ontologischen Grundtermtupel, mit denen die Stellen in M_0 markiert sind. „“

$I_w(M_0(\text{Konten}))$ = (0075417328,a_mueller,-900,-1000,0.2)
+(0083645173,w_meier,0,-22000,0.4)
+(0023321246,s_schulze,120000,-2000,0.2)
+(0023321246,w_meier,200,-1000,0.2);

für w = Konto Kunde Saldo Dispo Zinssatz

$I_v(M_0(\text{Kunde})$ = (a_mueller,21,„Schadowstr. 41, 45141 Essen“,{0075417328})
+(w_meier,32,„Obergath 5, 45147 Essen“,{0023321246,0083645173})
+(s_schulze,45,„Langenbeckstr. 73, 45141 Essen“,{0037458136});

für v = Kunde Alter Adresse MEN(Konto)

$I_h(M_0(\text{Kreditangebote})$ = $\varnothing_M$,

für h = Kunde Betrag.

Literaturverzeichnis

Vorbemerkungen:

- Alle Quellen im Literaturverzeichnis werden wie folgt aufgeführt: In der ersten Zeile wird der *Referenztitel* der Quelle angegeben. Er entspricht der Form, die im Text Verwendung findet, wenn auf die Quelle hingewiesen wird.
- Bei der Vergabe der Referenztitel wird bei *einem* Autor dessen Nachname, gefolgt von dem Erscheinungsjahr der Quelle in Klammern, verwendet. Bei *zwei* Autoren werden beide getrennt von einem Schrägstrich („/“) aufgeführt. Bei *mindestens drei* Autoren wird der erste Autor mit dem Zusatz „et al.“ („und andere“) aufgeführt.
- Zu *Internetquellen* wird die dafür Verantwortliche Instanz aufgeführt. Dies können sowohl natürliche als auch juristische Personen sein. Zu den Internetquellen wird die zum Aufrufdatum gültige Internetadresse (URL) angegeben. Gegebenenfalls nicht mehr abrufbare Seiten können vom Autor in einer Offline-Version erfragt werden.

ABECKER/VAN ELST (2004)
ABECKER, A.; VAN ELST, L.: Ontologies for Knowledge Management. In: STAAB, S.; STUDER, R. (Hrsg.): Handbook on Ontologies. Berlin et al. 2004, S. 435-454.

ABITEBOUL/GRUMBACH (1991)
ABITEBOUL, S.; GRUMBACH, S.: A rule-based language with functions and sets. In: ACM Transactions on Database Systems (TODS), 16. Jg. (1991), Nr. 1, S. 1-30.

ADAM ET AL. (1998)
ADAM, N. R.; ATLURI, V.; HUANG, W.: Modeling and Analysis of Workflows Using Petri Nets. In: Journal of Intelligent Information Systems, 10. Jg. (1998), Nr. 2, S. 131-158.

AITKEN/CURTIS (2002)
AITKEN, S.; CURTIS, J.: A Process Ontology. In: GÓMEZ-PÉREZ, A.; BENJAMINS, R. (Hrsg.): Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 13th International Conference, EKAW 2002, Siguenza, Spain, October 1-4, 2002, Proceedings. Berlin et al. 2002, S. 108-113.

ALAN (2002)
ALAN, Y.: Methoden zur Akquisition von Wissen über Kompetenzen. Projektbericht 2/2002, Projekt KOWIEN, Institut für Produktion und Industrielles Informationsmanagement, Universität Essen. Essen 2002.

ALBERT (2000)
ALBERT, H.: Kritischer Rationalismus. Tübingen 2000.

AMBLE (1987)
AMBLE, T.: Logic Programming and Knowledge Engineering. Wokingham et al. 1987.

ANGELE/LAUSEN (2004)
ANGELE, J.; LAUSEN, G.: Ontologies in F-Logic. In: STAAB, S.; STUDER, R. (Hrsg.): Handbook on Ontologies. Berlin et al. 2004, S. 29-50.

ANTONIOU (1996)
ANTONIOU, G.: Nonmonotonic Reasoning. Cambridge – London 1996.

ANTONIOU/VAN HARMELEN (2004)
ANTONIOU, G.; VAN HARMELEN, F.: A Semantic Web Primer. Cambridge 2004.

AOUMEUR (2001)
AOUMEUR, N.: Specifying and Validating Consistent and Dynamically Evolving Concurrent Information Systems: An Object Petri-net Based Approach. Dissertation, Universität Magdeburg. Magdeburg 2001.

AOUMEUR/SAAKE (2002)
AOUMEUR, N.; SAAKE, G.: A component-based Petri net model for specifying and validating cooperative information systems. In: Data & Knowledge Engineering, 42. Jg. (2002), Nr. 2, S. 143-187.

ASHBY (1974)
ASHBY, W.R.: Einführung in die Kybernetik. Frankfurt a.M. 1974.

BAADER ET AL. (2004)
BAADER, F.; HORROCKS, I.; SATTLER, U.: Description Logics. In: STAAB, S.; STUDER, R. (Hrsg.): Handbook on Ontologies. Berlin et al. 2004, S. 3-28.

BAADER/NUTT (2002)
BAADER, F.; NUTT, W.: Basic Description Logics. In: BAADER, F.; CALVANESE, D.; MCGUINESS, D.L.; NARDI, D.; PATEL-SCHNEIDER, P.F. (Hrsg.): The Description Logic Handbook. Cambridge 2002. S. 47-100.

BACLAWSKI ET AL. (2002)
BACLAWSKI, K.; KOKAR, M.K.; KOGUT, P.A.; HART, L.; SMITH, J.; LETKOWSKI, J.; EMERY, P.: Extending the Unified Modeling Language for ontology development. In: Software and System Modeling, 1. Jg. (2002), Nr. 2, S. 142-156.

BALDAN ET AL. (2000)
BALDAN, P.; BUSI, N.; CORRADINI, A.; PINNA, M.G.: Functorial Concurrent Semantics for Petri Nets with Read and Inhibitor Arcs. In: PALAMIDESSI, C. (Hrsg.): CONCUR 2000 – Concurrency Theory, 11th International Conference, University Park, PA, USA, August 22-25, 2000, Proceedings. Berlin et al. 2000, S. 442-457.

BALDUZZI ET AL. (2001)
BALDUZZI, F.; GIUA, A.; SEATZU, C.: Modelling and simulation of manufacturing systems with first-order hybrid Petri nets. In: International Journal of Production Research, 39. Jg. (2001), Nr. 2, S. 255-282.

BALLING (1997)
BALLING, R.: Kooperation: strategische Allianzen, Netzwerke, Joint Ventures und andere Organisationsformen zischenbetrieblicher Zusammenarbeit in Theorie und Praxis. Frankfurt a.M. 1997.

BARROS/TER HOFSTEDE (1998)
BARROS, A.P.; TER HOFSTEDE, A.H.M.: Towards the construction of workflow-suitable conceptual modeling techniques. In: Information Systems Journal, 8. Jg. (1998), Nr. 4, S. 313-337.

BAUMGARTEN (1996)
BAUMGARTEN, B.: Petri-Netze. 2. Aufl., Mannheim et al. 1996.

BECHHOFER ET AL. (2004)
BECHHOFER, S.; VAN HARMELEN, F.; HENDLER, J.; HORROCKS, I.; MCGUINESS, D.L.; PATEL-SCHNEIDER, P.F.; STEIN, L.A.: OWL Web Ontology Language Reference – W3C

Recommendation 10 February 2004. Im Internet unter der URL: http://www.w3.org/TR/owl-ref/, Aufruf: 12.04.2005.

BECKER (1999)
BECKER, O.: Synergien und Konflikte zwischen objektorientierter und kybernetischer Modellierung. Dissertation, Universität Marburg. Frankfurt a.M. 1999.

BECKER ET AL. (1999)
BECKER, J.; ROSEMANN, M.; SCHÜTTE, R. (Hrsg.): Referenzmodellierung. Heidelberg 1999.

BECKETT/MCBRIDE (2004)
BECKETT, D.; MCBRIDE, B.: RDF/XML Syntax Specification. W3C Recommendation 10 February 2004. Im Internet unter der URL: http://www.w3.org/TR/rdf-syntax-grammar/, Aufruf: 12.04.2005.

BEIERLE ET AL. (1993)
BEIERLE, C.; HEDTSTÜCK, U.; PLETAT, U.; SIEKMANN, J.: An Order-sorted Predicate Logic with Closely Coupled Taxonomic Information. In: MEINKE, K.; TUCKER, J.V. (Hrsg.): Many-sorted Logic and its Applications. Chichester et al. 1993, S. 179-211.

BEIERLE/KERN-ISBERNER (2000)
BEIERLE, C.; KERN-ISBERNER, G.: Methoden wissensbasierter Systeme. Wiesbaden 2000.

BEISEL (1996)
BEISEL, R.: Synergetik und Organisationsentwicklung. Dissertation, Universität München. 2. Aufl., Mering 1996.

BENCH-CAPON ET AL. (2003)
BENCH-CAPON, T.; MALCOLM, G.; SHAVE, M.: Semantics for Interoperability: Relating Ontologies and Schemata. In: MARÍK, V.; RETSCHITZEGGER, W.; STEPÁNKOVÁ, O. (Hrsg.): Database and Expert Systems Applications. 14th International Conference, DEXA 2003, Prague, Czech Republic, September 1-5, 2003, Proceedings. Berlin et al. 2003, S. 703-712.

BENCH-CAPON/MALCOLM (1999)
BENCH-CAPON, T.; MALCOLM, G.: Formalising Ontologies and Their Relations. In: BENCH-CAPON, T.; SODA, G.; MIN TJOA, A. (Hrsg.): Database and Expert Systems Applications. 10th International Conference, DEXA '99, Florence, Italy, August 30 – September 3, 1999, Proceedings. Berlin et al. 1999, S. 250-259.

BENSLIMANE ET AL. (2002)
BENSLIMANE, D.; HACID, M.-S.; TERZI, E.; TOUMANI, F.: A Class-Based Logic Language for Ontologies. In: ANDREASEN, T.; MOTRO, A.; CHRISTIANSEN, H.; LARSEN, H. L. (Hrsg.): Flexible Query Answering Systems, 5th International Conference, FQAS 2002, Copenhagen, Denmark, October 27-29, 2002, Proceedings. Berlin et al. 2002, S. 56-70.

BERGAMASCHI/SARTORI (1992)
BERGAMASCHI, S.; SARTORI, C.: On Taxonomic Reasoning in Conceptual Design. In: ACM Transactions on Database Systems (TODS), 17. Jg. (1992), Nr. 3, S. 385-422.

BERNERS-LEE ET AL. (1998)
BERNERS-LEE, T.; FIELDING, R.; IRVINE, U.C.; MASINTER, L.: Uniform Resource Identifieres (URI): Generic Syntax. Im Internet unter der URL: http://www.ietf.org/rfc/rfc2396.txt, Aufruf: 12.04.2005.

BERSTEL/BOASSON (2002)
BERSTEL, J.; BOASSON, L.: Formal Properties of XML grammars and languages. In: Acta Informatica, 38. Jg. (2002), Nr. 9, S. 649-671.

BERTOLAZZI ET AL. (2001)
BERTOLAZZI, P.; KRUSICH, C.; MISSIKOFF, M.: An Approach to the Defintion of a Core Enterprise Ontology: CEO. In: O. HRSG.: International Workshop on Open Enterprise Solutions: Systems, Experiences, and Organizations. Proceedings im Internet unter der URL: http://cersi.luiss.it/oesseo2001/papers/papers.htm, Aufruf: 12.04.2005.

BERTRAM (1996)
BERTRAM, M.: Das Unternehmensmodell als Basis der Wiederverwendung bei der Geschäftsprozeßmodellierung. In: VOSSEN, J.; BECKER, J. (Hrsg.): Geschäftsprozeßmodellierung und Workflow-Management. Bonn et al. 1996, S. 81-101.

BEST (1995)
BEST, E.: Semantik – Theorie sequentieller und paralleler Programmierung. Braunschweig – Wiesbaden 1995.

BIBEL (1992)
BIBEL, W.: Deduktion. München – Wien 1992.

BIBEL (1993)
BIBEL, W.: Wissensrepräsentation und Inferenz. Braunschweig – Wiesbaden 1993.

BIBERSTEIN ET AL. (2001)
BIBERSTEIN, O.; BUCHS, D.; GUELFI, N.: Object-Oriented Nets with Algebraic Specifications: The CO-OPN/2 Formalism. Aus: AGHA, G.; DE CINDIO, F.; ROZENBERG, G. (Hrsg.): Concurrent Object-Oriented Programming and Petri Nets, Advances in Petri Nets. Berlin et al. 2001. S. 73-130.

BOBEANU ET AL. (2004)
BOBEANU, C.-V.; KERCKHOFFS, E.J.H.; VAN LANDEGHEM, H.: Modeling of Discrete Event Systems: A Holistic and Incremental Approach Using Petri Nets. In: ACM Transactions on Modeling and Computer Simulation (TOMACS), 14. Jg. (2004), Nr. 4, S. 389-423.

BOMAN ET AL. (1997)
BOMAN, M.; BUBENKO JR., J.A.; JOHANNNESSON, P.; WANGLER, B.: Conceptual Modeling. London et al. 1997.

BORST/AKKERMANS (1997)
BORST, P.; AKKERMANS, H.: Engineering Ontologies. In: International Journal of Human-Computer Studies, 46. Jg. (1997), Nr. 2-3, S. 365-406.

BOSSEL (1994)
BOSSEL, H.: Modellbildung und Simulation. 2. Aufl., Braunschweig- Wiesbaden 1994.

BOWDEN (2000)
BOWDEN, F.D.J.: A brief survey and synthesis of the roles of time in Petri nets. In: Mathematical and Computer Modeling, 31. Jg. (2000), Nr. 10-12, S. 55-68.

BRANDSTÄDT (1994)
BRANDSTÄDT, A.: Graphen und Algorithmen. Stuttgart 1994.

BRAY ET AL. (2004)
BRAY, T.; PAOLI, J.; SPERBERG-MCQUEEN, C.M.; MALER, E.; YERGEAU, F.: Extensible Markup Language (XML) 1.0 (Third Edition). W3C Recommendation 04 February

2004.
Im Internet unter der URL: http://www.w3.org/TR/REC-xml/, Aufruf: 12.04.2005.

BRICKLEY/GUHA (2004)
BRICKLEY, D.; GUHA, R.V.: RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation 10 February 2004. Im Internet unter der URL: http://www.w3.org/TR/rdf-schema/, Aufruf: 12.04.2005.

BUDIN (1996)
BUDIN, G.: Wissensorganisation und Terminologie. Tübingen 1996.

CANN (1993)
CANN, R.: Formal Semantics. New York – Victoria 1993.

CARDELLI/WEGNER (1985)
CARDELLI, L.; WEGNER, P.: On Understanding Types, Data Abtstraction, and Polymorphism. In: Computing Surveys, 17. Jg. (1985), Nr. 4, S. 471-522.

CARSTENSEN ET AL. (2001)
CARSTENSEN, K.-U.; EBERT, C.; ENDRISS, C.; JEKAT, S.; KLABUNDE, R.; LANGER, H.: Computerlinguistik und Sprachtechnologie. Heidelberg 2001.

CHRISTENSEN/HANSEN (1993)
CHRISTENSEN, S.; HANSEN, N.D.: Coloured Petri Nets Extended with Place Capacities, Test Arcs and Inhibitor Arcs. In: MARSAN, M.A. (Hrsg.): Application and Theory of Petri Nets 1993, 14th International Conference, Chicago, Illinois, USA, June 21-25, 1993, Proceedings. Berlin et al. 1993, S. 186-205.

CIARDO/ZIAL (1996)
CIARDO, G.; ZIAL, R.: Well-Defined Stochastic Petri-Nets. In: O. HRSG.: MASCOTS '96, Proceedings of the Fourth International Workshop on Modeling, Analysis, and Simulation On Computer and Telecommunication Systems, February 1-3, 1996, San Jose, California, USA. San Jose 1996, S. 278-284.

COENEN/KLAPSING (2001)
COENEN, W.; KLAPSING, R.: Utilizing Host-Formalisms to Extend RDF-Semantics. In: CRUZ, I.F.; DECKER, S.; EUZENAT, J.; MCGUINNESS, D.L. (Hrsg.): Proceedings of SWWS'01, The first Semantic Web Working Symposium, Stanford University, California, USA, July 30 - August 1, 2001. Im Internet unter der URL: http://www.semanticweb.org/SWWS/program/full/SWWSProceedings.pdf. S. 181-193, Aufruf: 12.04.2005.

CONNOLLY ET AL. (2001)
CONNOLLY, D.; VAN HARMELEN, F.; HORROCKS, I.; MCGUINESS, D.L.; PATEL-SCHNEIDER, P.F.; STEIN, L.A.: DAML+OIL (March 2001) Reference Description – W3C Note 18 December 2001. Im Internet unter der URL: http://www.w3.org/TR/daml+oil-reference, Aufruf: 12.04.2005.

CORBY ET AL. (2000)
CORBY, O.; DIENG, R.; HÉBERT, C.: A Conceptual Graph Model for W3C Resource Description Framework. In: GANTER, B.; MINEAU, G.W. (Hrsg.): Conceptual Structures: Logical, Linguistic, and Computational Issues, 8th International Conference on Conceptual Structures, ICCS 2000, Darmstadt, Germany, August 14-18, 2000, Proceedings. Berlin et al. 2000, S. 468-482.

CORCHO/GÓMZ-PÉREZ (2000)
CORCHO, O.; GÓMEZ-PÉREZ, G.: A Roadmap to Ontology Specification Languages. In: DIENG, R.; CORBY, O. (Hrsg.): Knowledge Acquisition, Modeling and Management. 12th International Conference, EKAW 2000, Juan-les-Pins, France, October 2-6, 2000. Berlin et al. 2000, S. 80-96.

CRANEFIELD ET AL. (2003)
CRANEFIELD, S.; PAN, J.; PURVIS, M.K.: A UML ontology and derived content language for a travel booking scenario. In: CRANEFIELD, S.; FININ, T.W.; TAMMA, V.A.M.; WILLMOTT, S. (Hrsg.): Ontologies in Agent Systems 2003, Proceedings of the Workshop on Ontologies in Agent Systems (OAS 2003) at the 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems, Melbourne, Australia, July 15, 2003. Melbourne 2003, S. 55-62.

CROW/SHADBOLT (2001)
CROW, L.; SHADBOLT, N.: Extracting focused knowledge from the semantic web. In: International Journal of Human-Computer Studies, 54. Jg. (2001), Nr. 1, S. 155-184.

DAHR (1994)
DAHR, M.: Petri Net Semantics of Logic Programs and Deductive Databases. Hamburg 1994.

DAS (1992)
DAS, S.K.: Deductive Databases and Logic Programming. Wokingham et al. 1992.

DAVID/ALLA (2001)
DAVID, R.; ALLA, H.: On Hybrid Petri Nets. In: Discrete Event Dynamic Systems: Theory and Applications, 11. Jg. (2001), Nr. 1-2, S. 9-40.

DE MAINDREVILLE/SIMON (1987)
DE MAINDREVILLE, C.; SIMON, E.: A Predicate Transition Net for Evaluating Queries against Rules in a DBMS. Rapports de Recherche, Nr. 604, Institut National de Recherche en Informatique et en Automatique. Le Chesney Cedex 1987.

DE MICHELIS ET AL. (1997)
DE MICHELIS, G.; DUBOIS, E.; JARKE, M.; MATTHES, F.; MYLOPOULOS, J.; PAPAZOGLOU, M.; POHL, K.; SCHMIDT, J.; WOO, C.; YU, E.: Cooperative Information Systems: A Manifesto. Beitrag zur 4th International Conference on Cooperative Information Systems, June 1996. Im Internet unter der URL:
http://www.sts.tu-harburg.de/papers/1997/DDJ+97.pdf, Aufruf: 11.04.2005.

DECKER (2002)
DECKER, S.: Semantic Web Methods for Knowledge Management. Disseratation, Universität Karlsruhe. Karlsruhe 2002.

DESEL/JUHÁS (2000)
DESEL, J.; JUHÁS, G.: "What is a Petri Net?" In: EHRIG, H.; JUHÁS, G.; PADBERG, J.; ROZENBERG, G. (Hrsg.): Unifying Petri Nets, Advances in Petri Nets. Berlin et al. 2001, S. 1-25.

DESEL/REISIG (1998)
DESEL, J.; REISIG, W.: Place/Transition Petri Nets. In: REISIG, W.; ROZENBERG, G. (Hrsg.): Lectures on Petri Nets I: Basic Models. Berlin et al. 1998, S. 122-173.

DIESTEL (2000)
DIESTEL, R.: Graphentheorie. Elektronische Ausgabe 2000. Berlin et al. 2000.

DING ET AL. (2002)
DING, Y.; FENSEL, D.; KLEIN, M.; OMELAYENKO, B.: The semantic web: yet another hip? In: Data & Knowledge Engineering, 41. Jg. (2002), Nr. 2-3, S. 205-227.

DING ET AL. (2004)
DING, Y.; FENSEL, D.; KLEIN, M.; OMELAYENKO, B.; SCHULTEN, E.: The role of Ontologies in eCommerce. In: STAAB, S.; STUDER, R. (Hrsg.): Handbook on Ontologies. Berlin et al. 2004, S. 593-615.

DITTRICH/GATZIU (2000)
DITTRICH, K.R.; GATZIU, S.: Aktive Datenbanksysteme. 2. Aufl., Heidelberg 2000.

DOMINGUE ET AL. (1999)
DOMINGUE, J.; MOTTA, E.; GARCIA, O.C.: Knowledge Modelling in WebOnto and OCML – A User Guide. Im Internet unter der URL:
http://kmi.open.ac.uk/projects/ocml/ocml-webonto-guide.zip, Aufruf: 12.04.2005.

DOVIER ET AL. (2000)
DOVIER, A.; PIAZZA, C.; PONTELLI, E.; ROSSI, G.: Sets and Constraint Logic Programming. In: ACM Transactions on Programming Languages and Systems (TOPLAS), 22. Jg. (2000), Nr. 5, S. 861-931.

DRESBACH (1999)
DRESBACH, S.: Epistemologische Überlegungen zu Modellen in der Wirtschaftsinformatik. In: BECKER, J.; KÖNIG, W.; SCHÜTTE, R.; WENDT, O.; ZELEWSKI, S. (Hrsg.): Wirtschaftsinformatik und Wissenschaftstheorie. Wiesbaden 1997, S. 71-94.

DUSSART ET AL. (2002)
DUSSART, A.; AUBERT, B.A.; PATRY, M.: An Evaluation of Inter-Organizational Workflow Modelling Formalisms. Scientific Series, CIRANO – Centre interuniversitaire de recherche en analyse des organisations. Im Internet unter der URL:
http://www.cirano.qc.ca/pdf/publication/2002s-64.pdf, Aufruf: 12.04.2005.

EBBINGHAUS ET AL. (1992)
EBBINGAHAUS, H.D.; FLUM, J.; THOMAS, W.: Einführung in die mathematische Logik. 3. Aufl., Mannheim et al. 1992.

ECO (1988)
ECO, U.: Wer ist Schuld an der Konfusion von Denotation und Bedeutung? Versuch einer Spurensicherung. In: Zeitschrift für Semiotik, 10. Jg. (1988), Nr. 3, S. 189-207.

EHRICH ET AL. (1989)
EHRICH, H.-D.; GOGOLLA, M.; LIPECK, U. W.: Algebraische Spezifikation abstrakter Datentypen. Stuttgart 1989.

EHRIG ET AL. (1999)
EHRIG, H.; MAHR, B.; CORNELIUS, F.; GROßE-RHODE, M.; ZEITZ, P.: Mathematisch-strukturelle Grundlagen der Informatik. Berlin et al. 1999.

EIRUND ET AL. (2000)
EIRUND, H.; MÜLLER, B.; SCHREIBER, G.: Formale Beschreibungsverfahren der Informatik. Leipzig – Wiesbaden 2000.

ERDMANN (2001)
ERDMANN, M.: Ontologien zur konzeptuellen Modellierung der Semantik von XML. Dissertation, Universität Karlsruhe. Norderstedt 2001.

ERK/PRIESE (2002)
ERK, K.; PRIESE, L.: Theoretische Informatik. 2. Aufl., Berlin et al. 2002.

ERWIN (2002)
ERWIN, T.: Entwurf von Geschäftsprozessen mit Petri-Netzen. Dissertation, Universität Karlsruhe. Karlsruhe 2002.

FARQUHAR ET AL. (1997)
FARQUHAR, A.; FIKES, R.; RICE, J.: The Ontolingua Server: A Tool for Collaborative Ontology Construction. In: International Journal of Human-Computer Studies, 46. Jg. (1997), Nr. 6, S. 707-727.

FENSEL (2000)
FENSEL, D.: Relating Ontology Languages and Web Standards. In: EBERT, J.; FRANK, U. (Hrsg.): Modelle und Modellierungssprachen in Informatik und Wirtschaftsinformatik. Beiträge des Workshops "Modellierung 2000", St. Goar, 5.-7. April 2000. Koblenz 2000, S. 111-128.

FENSEL (2001)
FENSEL, D.: Ontologies and Electronic Commerce. In: IEEE Intelligent Systems, 16. Jg. (2001), Nr. 1, S. 8-14.

FENSEL (2001A)
FENSEL, D.: Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. Berlin et al. 2001.

FERSTL/SINZ (1998)
FERSTL, O.K.; SINZ, E.J.: Grundlagen der Wirtschaftsinformatik. 3. Aufl., München – Wien 1998.

FETTKE/LOOS (2004)
FETTKE, P.; LOOS, P.: Referenzmodellierungsforschung. In: Wirtschaftsinformatik, 46. Jg. (2004), Nr. 5, S. 331-340.

FONTANARI (1996)
FONTANARI, M.: Kooperationsgestaltungsprozesse in Theorie und Praxis. Dissertation, Universität Trier. Berlin 1996.

FOWLER (2004)
FOWLER, M.: UML Distilled. Boston et al. 2004.

FOX/GRUENINGER (1997)
FOX, M.S.; GRUENINGER, M.: Ontologies for Enterprise Modeling. In: KOSANKE, K.; NELL, J.G. (Hrsg.): Enterprise Engineering and Integration: Building International Consensus. Berlin et al. 1997, S. 190-220.

FRANK (1994)
FRANK, U.: Multiperspektivische Unternehmensmodellierung. Habilitationsschrift, Universität Marburg. Marburg 1994.

FRANK (1999)
FRANK, U.: Zur Verwendung formaler Sprachen in der Wirtschaftsinformatik: Ein notwendiges Merkmal eines wissenschaftlichen Anspruchs oder Ausdruck eines übertriebenen Szientizismus? In: BECKER, J.; ROSEMANN, M.; SCHÜTTE, R. (Hrsg.): Referenzmodellierung. Heidelberg 1999, S. 127-158.

FRANK (2000)
FRANK, U.: Modelle als Evaluationsobjekt. In: HEINRICH, L.J.; HÄNTSCHEL, I. (Hrsg.): Evaluation und Evaluationsforschung in der Wirtschaftsinformatik. München – Wien 2000, S. 339-352.

FRANK (2003)
FRANK, U.: Ebenen der Abstraktion und ihre Abbildung auf konzeptionelle Modelle – oder: Anmerkungen zur Semantik von Spezialisierungs- und Instanzierungsbeziehungen. In: Emisa-Forum, 23. Jg. (2003), Nr. 2, S. 14-18.

FRANK/VAN LAAK (2003)
FRANK, U.; VAN LAAK, B.L.: Anforderungen an Sprachen zur Modellierung von Geschäftsprozessen. Arbeitsbericht Nr. 34 des Instituts für Wirtschaftsinformatik, Universität Koblenz – Landau. Koblenz 2003.

FREGE (1966)
FREGE, G.: Logische Untersuchungen. Göttingen 1966.

FREGE (1969)
FREGE, G.: Über Sinn und Bedeutung. In: PATZIG, G. (Hrsg.): Funktion, Begriff, Bedeutung – Fünf logische Studien. 3. Aufl., Göttingen 1969, S. 18-39.

GALLIER (1986)
GALLIER, J.H.: Logic for Computer Science. New York et al. 1986.

GANTER/WILLE (1996)
GANTER, B.; WILLE, R.: Formale Begriffsanalyse. Berlin et al. 1996.

GENESERETH/FIKES (1997)
GENSERETH, M.R.; FIKES, R.E.: Knowledge Interchange Format (Version 3.0). Reference Manual. Im Internet unter der URL: http://logic.stanford.edu/kif/Hypertext/kif-manual.html, Aufruf: 12.04.2005.

GENRICH (1987)
GENRICH, H.J.: Predicate/Transition Nets. In: BRAUER, W.; REISIG, W.; ROZENBERG, G. (Hrsg.): Petri Nets: Central Models and Their Properties, Advances in Petri Nets 1986, Part I, Proceedings of an Advanced Course, Bad Honnef, 8.-19. September 1986. Berlin et al. 1987, S. 207-247.

GENRICH (2002)
GENRICH, H.J.: Dynamical Quantities in Net Systems. In: Formal Aspects of Computing, 14. Jg. (2002), Nr. 1, S. 55-89.

GEROGIANNIS ET AL. (1998)
GEROGIANNIS, V.C.; KAMEAS, A.D.; PINTELAS, P.E.: Comparative study and categorization of high-level petri-nets. In: The Journal of Systems and Software, 43. Jg. (1998), Nr. 2, S. 133-160.

GIRAULT/VALK (2003)
GIRAULT, C.; VALK, R.: Petri Nets for Systems Engineering. Berlin et al. 2003.

GLASERFELD (1987)
GLASERFELD, E.v.: Siegener Gespräche über radikalen Kosntruktivismus. In: SCHMIDT, S. (Hrsg.): Der Diskurs des radikalen Konstruktivismus. Frankfurt a.M. 1987, S. 401-440.

GODFREY ET AL. (1998)
GODFREY, P.; GRANT, J.; GRYZ, J.; MINKER, J.: Integrity Constraints: Semantics and Applications. In: CHOMICKI, J.; SAAKE, G. (Hrsg.): Logics for Databases and Information Systems. Boston et al. 1998, S. 265-306.

GOGUEN/MESEGUER (1992)
GOGUEN, J.A.; MESEGUER, J.: Order-Sorted Algebra I: Equational Deduction for Multiple Inheritance, Overloading, Exceptions and Partial Operations. In: Theoretical Computer Science, 105. Jg. (1992), Nr. 2, S. 217-273.

GÓMEZ-PÉREZ (2001)
GÓMEZ-PÉREZ, A.: Evaluation of Ontologies. In: International Journal of Intelligent Systems, 16. Jg. (2001), Nr. 3, S. 391-409.

GÓMEZ-PÉREZ (2004)
GÓMEZ-PÉREZ, A.: Ontology Evaluation. In: STAAB, S.; STUDER, R. (Hrsg.): Handbook on Ontologies. Berlin et al. 2004, S. 251-273.

GÓMEZ-PÉREZ ET AL. (2004)
GÓMEZ-PÉREZ, A.; FERNÁNDEZ-LÓPEZ, M.; CORCHO, O.: Ontological Engineering. Berlin et al. 2004.

GROSSO ET AL. (1997)
GROSSO, W.; GENNARI, J.; FERGERSON, R.; MUSEN, M.: When Knowledge Models Collide (How it Happens and What to Do). SMI Report No. SMI-97-0696. Im Internet unter der URL: http://smi-web.stanford.edu/pubs/SMI_Abstracts/SMI-97-0696.html, Aufruf: 12.04.2005.

GRUBER (1992)
GRUBER, T.: Ontolingua: A Mechanism to Support Portable Ontologies. Technical Report No. KSL-91-66, Knowledge Systems Laboratory, Stanford University. Im Internet unter der URL: ftp://ftp.ksl.stanford.edu/pub/KSL_Reports/KSL-91-66.ps.gz, Aufruf: 12.04.2005.

GRUBER (1992A)
GRUBER, T.: Toward principles for the Design of Ontologies used for Knowledge Sharing. Technical Report No. KSL-93-04, Knowledge Systems Laboratory, Stanford University. Im Internet unter der URL: ftp://ftp.ksl.stanford.edu/pub/KSL_Reports/KSL-93-04.ps.gz, Aufruf: 12.04.2005.

GRUBER (1993)
GRUBER, T.: A translation approach to portable ontology specifications. In: Knowledge Acquisition, 5. Jg. (1993), Nr. 2, S. 199-220.

GRUENINGER (2004)
GRUENINGER, M.: Ontology of the Process Specification Language. In: STAAB, S.; STUDER, R. (Hrsg.): Handbook on Ontologies. Berlin et al. 2004, S. 575-592.

GRUHN (1996)
GRUHN, V.: Geschäftsprozeß-Management als Grundlage der Software-Entwicklung. In: Informatik Forschung und Entwicklung, 11. Jg. (1996), Nr. 2, S. 94-101.

GRUHN/KAMPMANN (1996)
GRUHN, V.; KAMPMANN, M.: Modellierung unternehmensübergreifender Geschäftsprozesse mit FUNSOFT-Netzen. In: Wirtschaftsinformatik, 38. Jg. (1996), Nr. 4, S. 383-390.

GRUNINGER (1997)
GRUNINGER, M.: Integrated Ontologies for Enterprise Modelling. In: KOSANKE, K.; NELL, J.G. (Hrsg.): Enterprise Engineering and Integration: Building International Consensus. Berlin et al. 1997, S. 368-377.

GRUNINGER ET AL. (2000)
GUNINGER, M.; ATEFI, K.; FOX, M.S.: Ontologies to Support Process Integration in Enterprise Engineering. In: Computational & Mathematical Organization Theory, 6. Jg. (2000), Nr. 4, S. 381-394.

GUARINO (1995)
GUARINO, N.: Formal ontology, conceptual analysis and knowledge representation. In: International Journal of Human-Computer Studies, 43. Jg. (1995), Nr. 5-6, S. 625-640.

GUARINO (1997)
GUARINO, N.: Understanding, building and using ontologies. In: International Journal of Human-Computer Studies, 46. Jg. (1997), Nr. 2, S. 293-310.

GUARINO (1999)
GUARINO, N.: The Role of Identity Conditions in Ontology Design. In: FREKSA, C. (Hrsg.): Spatial information theory: cognitive and computational foundations of geographic information science. International Conference COSIT '99, Stade, Germany, August 25-29, 1999, Proceedings. Berlin et al. 1999, S. 221-234.

GUARINO/GIARETTA (1995)
GUARINO, N.; GIARETTA, P.: Ontologies and Knowledge Bases – Towards a Terminological Clarification. In: MARS, N.J.I. (Hrsg.): Towards very large knowledge bases: knowledge building & knowledge sharing. Amsterdam 1995, S. 25-32.

GUARINO/WELTY (2000)
GUARINO, N.; WELTY, C.: A Formal Ontology of Properties. In: DIENG, R.; CORBY, O. (Hrsg.): Knowledge Acquisition, Modeling and Management. 12th International Conference, EKAW 2000, Juan-les-Pins, France, October 2-6, 2000. Berlin et al. 2000, S. 97-112.

GUARINO/WELTY (2002)
GUARINO, N.; WELTY, C.: Evaluating ontological decisions with Ontoclean. In: Communications of the ACM, 45. Jg. (2002), Nr. 2, S. 61-65.

GUARINO/WELTY (2004)
GUARINO, N.; WELTY, C.A.: An Overview of OntoClean. In: STAAB, S.; STUDER, R. (Hrsg.): Handbook on Ontologies. Berlin et al. 2004, S. 151-171.

GUESSERIAN (1993)
GUESSERIAN, I.: Many-sorted Logic and Algebraic Semantics. In: MEINKE, K.; TUCKER, J.V. (Hrsg.): Many-sorted Logic and its Applications. Chichester et al. 1993, S. 123-134.

GUIZZARDI ET AL. (2002)
GUIZZARDI, G.; HERRE, H.; WAGNER, G.: On the General Ontological Foundations of Counceptual Modeling. In: SPACCAPIETRA , S.; MARCH, S.T.; KAMBAYASHI, Y. (Hrsg.): Conceptual Modeling – ER 2002. 21st International Conference on Conceptual Modeling, Tampere, Finland, October 7-11, 2002, Proceedings. Berlin et al. 2002, S. 65-78.

HAHN/SCHULZ (2004)
HAHN, U.; SCHULZ, S.: Building a Very Large Ontology from Medical Thesauri. In: STAAB, S.; STUDER, R. (Hrsg.): Handbook on Ontologies. Berlin et al. 2004, S. 133-150.

HAMEL/GOGUEN (1994)
HAMEL, L.H.; GOGUEN, J.A.: Towards a Provably Correct Compiler for OBJ3. In: HERMENEGILDO, M.V.; PENJAM, J. (Hrsg.): Programming Language Implementation and Logic Programming, 6th International Symposium, PLILP'94, Madrid, Spain, September 14-16, 1994, Proceedings. Berlin et al. 1994. S. 132-146.

HANSEN ET AL. (1992)
HANSEN, H.R.; MÜHLBACHER, R.; NEUMANN, G.: Begriffsbasierte Integration von Systemanalysemethoden. Heidelberg 1992.

HARRAS (2000)
HARRAS, G.: Concepts in Linguistics – Concepts in Natural Languag. In: GANTER, B.; MINEAU, G.W. (Hrsg.): Conceptual Structures: Logical, Linguistic, and Computational Issues, 8th International Conference on Conceptual Structures, ICCS 2000, Darmstadt, Germany, August 14-18, 2000, Proceedings. Berlin et al. 2000, S. 13-26.

HATZILYGEROUDIS/REICHGELT (1997)
HATZILYEROUDIS, I.; REICHGELT, H.: Handling inheritance in a system integrating logic in objects. In: Data & Knowledge Engineering, 21. Jg. (1997), Nr. 3, S. 253-280.

HE ET AL. (2003)
HE, X.; CHU, W.C.; YANG, H.: A new approach to verify rule-based systems using petri-nets. In: Information and Software Technology, 45. Jg. (2003), Nr. 10, S. 663-669.

HE ET AL. (2004)
HE, X.; YU, H.; SHI, T.; DING, J.; DENG, Y.: Formally analyzing software architectural specifications using SAM. In: The Journal of Systems and Software, 71. Jg. (2004), Nr. 1, S. 11-29.

HE/LEUNG (2002)
HE, M.; LEUNG, H.: Agents in E-Commerce: State of the Art. In: Knowledge and Information Systems, 4. Jg. (2002), Nr. 4, S. 257-282.

HEFLIN (2001)
HEFLIN, J.: Towards the Semantic Web – Knowledge Representation in a Dynamic, Distributed Environment. Dissertation, University of Maryland – USA. Maryland 2001.

HEFLIN ET AL. (1999)
HEFLIN, J.; HENDLER, J.; LUKE, S.: SHOE: A Knowledge Representation Language for Internet Applications. Technical Report CS-TR-4078 (UMIACS TR-99-71), Department of Computer Science, Universität Maryland, USA. Im Internet unter der URL: http://www.cs.umd.edu/projects/plus/SHOE/pubs/techrpt99.pdf, Aufruf: 12.04.2005.

HEFLIN/HENDLER (2001)
HEFLIN, J.; HENDLER, J.: A Portrait of the Semantic Web in Action. In: IEEE Intelligent Systems, 16. Jg. (2001), Nr. 2, S. 54-59.

HERMES (1991)
HERMES, H.: Einführung in die mathematische Logik. 5. Aufl., Stuttgart 1991.

HERBST (1997)
HERBST, H.: Business Rule-Oriented Conceptual Modeling. Heidelberg 1997.

HERRMANN ET AL. (1996)
HERRMANN, T.; GRABOWSKI, J.; SCHWEIZER, K.; GRAF, R.: Die mentale Repräsentation von Konzepten, Wörtern und Figuren. In: GRABOWSKI, J.; HARRAS, G.; HERRMANN, T. (Hrsg.): Bedeutung – Konzepte – Bedeutungskonzepte. Opladen 1996, S. 120-152.

HESSE (2002)
HESSE, W.: Ontologie(n). In: Informatik Spektrum, 25. Jg. (2002), Nr. 6, S. 477-480.

HJELM (2001)
HJELM, J.: Creating the Semantic Web with RDF. New York 2001.

HOPPENBROUWERS (1997)
HOPPENBROUWERS, J.: Conceptual Modeling and the Lexicon. Dissertation, Universität Brabant, Niederlande. Brabant 1997.

HORROCKS ET AL. (2003)
HORROCKS, I.; PATEL-SCHNEIDER, P.F.; VAN HARMELEN, F.: From SHIQ and RDF to OWL: the making of a Web Ontology Language. In: Web Semantics, 1. Jg. (2003), Nr. 1, S. 7-26.

IZUMI/YAMAGUCHI (2002)
IZUMI, N.; YAMAGUCHI, T.: Integration of heterogeneous repositories based on ontologies for EC applications development. In: Electronic Commerce Research and Applications, 1. Jg. (2002), S. 77-91.

JAESCHKE (1996)
JAESCHKE, P.: Integrierte Unternehmensmodellierung. Wiesbaden 1996.

JENSEN (1996)
JENSEN, K.: Coloured Petri Nets. Volume 1 – Basic Concepts. 2. Aufl., Berlin et al. 1996.

JOHNSON/LAIRD (1983)
JOHNSON-LAIRD, P.N.: Mental Models. Cambridge 1983.

JONES/PATON (1999)
JONES, D.M.; PATON, R.C.: Toward principles for the representation of hierarchical knowledge in formal ontologies. In: Data & Knowledge Engineering, 31. Jg. (1999), Nr. 2, S. 99-113.

JOUVE ET AL. (2003)
JOUVE, D.; AMGHAR, Y.; CHABBAT, B.; PINON, J.-M.: Conceptual framework for document semantic modelling: an application to document and knowledge management in the legal domain. In: Data & Knowledge Engineering, 46. Jg. (2003), Nr. 3, S. 345-375.

JUOPPERI (1995)
JUOPPERI, J.: PrT-Net Based Analysis of Information Flow Security Nets. Research Report No. 34, March 1995, Digital Systems Laboratory, Helsinki University of Technology. Helsinki 1995.

KANEIWA (2001)
KANEIWA, K.: An Order-sorted Logic with Predicate-Hierarchy, Eventuality and Implicit Negation. Dissertation, School of Information Science, Ishikawa – Japan. Ishikawa 2001.

KANEIWA (2004)
KANEIWA, K.: Order-sorted logic programming with predicate hierarchy. In: Artificial Intelligence, 158. Jg. (2004), Nr. 2, S. 155-188.

KARBACH/LINSTER (1990)
KARBACH, W.; LINSTER, M.: Wissensakquisition für Expertensysteme: Techniken, Modelle und Softwarewerkzeuge. München 1990.

KARP ET AL. (1999)
KARP, P.D.; CHAUDHRI, V.K.; THOMERE, J.: XOL: An XML-Based Ontology exchange Language. Im Internet unter der URL: http://www.ai.sri.com/~pkarp/xol/xol.html, Aufruf: 12.04.2005.

KARVOUNARAKIS ET AL. (2002)
KARVOUNARAKIS, G.; ALEXAKI, S.; CHRISTOPHIDES, V.; PLEXOUSAKIS, D.; SCHOLL, M.: RQL: a declarative query language for RDF. In: O. HRSG.: WWW 2002. Proceedings of the Eleventh International World Wide Web Conference, WWW2002, Honolulu, Hawaii, USA, May 7-11, 2002. New York 2002, S. 592-603.

KASCHEK (2004)
KASCHEK, R.: A little theory of abstraction. In: RUMPE, B.; HESSE, W. (Hrsg.): Modellierung 2004 – Proceedings zur Tagung 23.-26. März 2004, Marburg. Berlin et al. 2004, S. 75-92.

KHAN ET AL. (2004)
KHAN, L.; MCLEOD, D.; HOVY, E.H.: Retrieval effectiveness of an ontology-based model for information selection. In: The VLDB Journal, 13. Jg. (2004), Nr. 1, S. 71-85.

KIFER ET AL. (1995)
KIFER, M.; LAUSEN, G.; WU. J.: Logical Foundations of Object-Oriented and Frame-Based Languages. In: Journal of the ACM, 42. Jg. (1995), Nr. 4, S. 741-843.

KIM (2002)
KIM, H.: Predicting how ontologies for the semantic web will evolve. In: Communications of the ACM, 45. Jg. (2002), Nr. 2, S. 48-54.

KINDLER (1995)
KINDLER, E.: Modularer Entwurf verteilter Systeme mit Petrinetzen. Dissertation, Universität TU München. Berlin 1995.

KLAPSING (2003)
KLAPSING, R.: Beschreibung von Web-basierten Informationssystemen mittels RDF-Metadaten. Dissertation, Universität Essen. Essen 2003.

KLEIN (1996)
KLEIN, S.: Interorganisationssysteme und Unternehmensnetzwerke. Wiesbaden 1996.

KLEINJOHANN (1993)
KLEINJOHANN, E.: Integrierte Entwurfsberatung auf der Basis erweiterter Prädikat-Transitionsnetze. Dissertation, Universität-Gesamthochschule Paderborn. Paderborn 1993.

KLUSCH (1996)
KLUSCH, M.: Rational kooperative Erkennung von Interdatenbankabhängigkeiten. Dissertation, Universität Kiel. Kiel 1996.

KNOLMAYER ET AL. (2000)
KNOLMAYER, G.; ENDL, R.; PFAHRER, M: Modeling Processes and Workflows by Business Rules. In: VAN DER AALST, W.; DESEL, J.; OBERWEIS, A. (Hrsg.): Business Process Management. Models Techniques, and Empirical Studies. Berlin et al. 2000, S. 16-29.

KOGUT ET AL. (2002)
KOGUT, P.; CRANEFIELD, S.; HART, L.; DUTRA, M.; BACLAWSKI, K.; KOKAR, M.; SMITH, J.: UML for ontology development. In: The Knowledge Engineering Review, 17. Jg. (2002), Nr. 1, S. 61-64.

KOHLHASE (1992)
KOHLHASE, M.: Beweissysteme mit Logiken höherer Stufe. In: BLÄSIUS, K.H.; BÜRCKERT, H.-J. (Hrsg.): Deduktionssysteme. 2. Aufl., München – Wien 1992, S. 213-238.

KORCZYNSKI ET AL. (1990)
KORCZYNSKI, W.; DÜPMEIER, C.; SÜSS, W.: Eine Einführung in die Grundlagen der Theorie der Höheren Petri-Netze. Forschungsberichte Kernforschungs-Zentrum Karlsruhe Nr. 4636. Karlsruhe 1990.

KOUBARAKIS/PLEXOUSAKIS (2002)
KOUBARAKIS, M.; PLEXOUSAKIS, D.: A formal framework for business process modelling and design. In: Information Systems, 27. Jg. (2002), Nr. 5, S. 299-319.

KREOWSKI (1991)
KREOWSKI, H.-J.: Logische Grundlagen der Informatik. München – Wien 1991.

KÜHNE/STEIMANN (2004)
KÜHNE, T.; STEIMANN, F.: Tiefe Charakterisierung. In: RUMPE, B.; HESSE, W. (Hrsg.): Modellierung 2004 – Proceedings zur Tagung 23.-26. März 2004, Marburg. Berlin et al. 2004, S. 109-119.

KUPER (1987)
KUPER, G.M.: Logic Programming with Sets. In: O. HRSG.: Proceedings of the Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, March 23-25, 1987, San Diego, California. New York 1987, S. 11-20.

KUPER (1988)
KUPER, G.M.: On the Expressive Power of Logic Programming Languages with Sets. In: O. HRSG.: Proceedings of the Seventh ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, March 21-23, 1988, Austin, Texas. New York 1988, S. 10-14.

LASSILA/SWICK (1999)
LASSILA, O.; SWICK, R.R.: Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation 22 February 1999. Im Internet unter der URL: http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/, Aufruf: 12.04.2005.

LAUSEN ET AL. (1998)
LAUSEN, G.; LUDÄSCHER, B.; MAY, W.: On Active Databases: The Statelog Approach. In: FREITAG , B.; DECKER, H.; KIFER, M.; VORONKOV, A. (Hrsg.): Transactions and Change in Logic Databases. Berlin et al. 1998, S. 69-106.

LAUSEN/VOSSEN (1996)
LAUSEN, G.; VOSSEN, G.: Objekt-orientierte Datenbanken: Modelle und Sprachen. München – Wien 1996.

LAUTENBACH (2003)
LAUTENBACH, K.: Logical Reasoning and Petri Nets. In: VAN DER AALST, W.; BEST, E. (Hrsg.): Applications and Theory of Petri Nets 2003. 24th International Conference,

ICATPN 2003, Eindhoven, Niederlande, June 23-27, 2003, Proceedings. Berlin et al. 2003, S. 276-295.

LEHNER ET AL. (1995)
LEHNER, F.; HILDEBRAND, K.; MAIER, R.: Wirtschaftsinformatik. München – Wien 1995.

LEICH (2002)
LEICH, S.: Agentensoftware und Unternehmenskommunikation – Wahrnehmung und Beurteilung von Leistungen im E-Commerce. Wiesbaden 2002.

LENAT (1995)
LENAT, D.B.: CYC: A Large-Scale Investement in Knowledge Infrastructure. In: Communications of the ACM, 38. Jg. (1995), Nr. 11, S. 45-48.

LENAT ET AL. (1990)
LENAT, D.B.; GUHA, R.V.; PITTMAN, K.; PRATT, D.; SHEPHERD, M.: CYC: Toward Programming with Common Sense. In: Communications of the ACM, 33. Jg. (1990), Nr. 8, S. 30-49.

LENZ (2003)
LENZ, K.: Modellierung und Ausführung von E-Business-Prozessen mit XML-Netzen. Dissertation, Universität Frankfurt a.M. Berlin 2003.

LI (1994)
LI, L.: High-Level Petri Net Model of Logic Program with Negation. In: IEEE Transactions on Knowledge and Data Engineering , 6. Jg. (1994), Nr. 3, S. 382-395.

LIN/CHANSON (1998)
LIN, C.; CHANSON S.T.: Logical Inference of Clauses Based on Petri Net Models. In: International Journal of Intelligent Systems, 13. Jg. (1998), Nr. 9, S. 821-840.

LÖBNER (2003)
LÖBNER, S.: Semantik. Berlin – New York 2003.

LUTZEIER (1995)
LUTZEIER, P.R.: Lexikologie. Tübingen 1995.

MÄDCHE (2002)
MÄDCHE, A.: Ontology learning for the semantic Web. Dissertation, Universität Karlsruhe. Boston et al. 2002.

MÄDCHE ET AL. (2000)
MÄDCHE, A.; SCHNURR, H.-P.; STAAB, S.; STUDER, R.: Representation Language-Neutral Modeling of Ontologies. In: EBERT, J.; FRANK, U. (Hrsg.): Modelle und Modellierungssprachen in Informatik und Wirtschaftsinformatik. Beiträge des Workshops "Modellierung 2000", St. Goar, 5.-7. April 2000. Koblenz 2000, S. 129-142.

MÄDCHE ET AL. (2003)
MÄDCHE, A.; MOTIK, B.; STOJANOVIC, L.: Managing multiple and distributed ontologies on the Semantic Web. In: The VLDB Journal, 12. Jg. (2003), Nr. 4, S. 286-302.

MÄDCHE/STAAB (2001)
MÄDCHE, A.; STAAB, S.: Ontology Learning for the Semantic Web. In: IEEE Intelligent Systems, 16. Jg. (2001), Nr. 2, S. 72-79.

MAIER (1993)
MAIER, A.: Einbettung von Konzepthierarchien in ein deduktives Datenbanksystem. Dissertation, Universität Trier. Sankt Augustin 1993.

MANOLA/MILLER (2004)
MANOLA, F.; MILLER, E.: RDF Primer. Im Internet unter der URL: http://www.w3.org/TR/2004/REC-rdf-primer-20040210/, Aufruf: 12.04.2005.

MANZANO (1993)
MANZANO, M.: Introduction to Many-sorted Logic. In: MEINKE, K.; TUCKER, J.V. (Hrsg.): Many-sorted Logic and its Applications. Chichester et al. 1993, S. 3-86.

MARTIN ET AL. (2004)
MARTIN, D.; BURSTEIN, M.; HOBBS, J.; LASSILA, O.; MCDERMOTT, D.; MCILRAITH, S.; NARAYANAN, S.; PAOLUCCI, M.; PARSIA, B.; PAYNE, T.; SIRIN, E.; SRINIVASAN, N.; SYCARA, K.: OWL-S: Semantic Markup for Web Services. W3C Member Submission 22 November 2004. Im Internet unter der URL: http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/, Aufruf: 12.04.2005.

MATURANA (2001)
MATURANA, H.R.: Der Baum der Erkenntnis. München 2001.

MCDERMOTT/DOU (2002)
MCDERMOTT, D.; DOU, D.: Representing Disjunction and Quantifiers in RDF. In: HORROCKS, I.; HENDLER, J. (Hrsg.): The Semantic Web – ISWC 2002, First International Semantic Web Conference, Sardinia, Italy, June 9-12, 2002, Proceedings. Berlin et al. 2002, S. 250-263.

MCGUINESS/VAN HARMELEN (2004)
MCGUINESS, D.; VAN HARMELEN, F.: OWL Web Ontology Language – Overview. Im Internet unter der URL: http://www.w3.org/TR/2004/REC-owl-features-20040210/, Aufruf: 12.04.2005.

MENZEL (1997)
MENZEL, C.: Modeling Method Ontologies: A Foundation for Enterprise Model Integration. In: FARQUHAR, A.; GRUNINGER, M. (Hrsg.): Ontological Engineering. Menlo Park 1997, S. 73-83.

MERTENS ET AL. (1998)
MERTENS, P.; GRIESE, J.; EHRENBERG, D.: Virtuelle Unternehmen und Informationsverarbeitung. Berlin et al. 1998.

MESSER (1999)
MESSER, B.: Zur Interpretation formaler Geschäftsprozess- und Workflow-Modelle. In: BECKER, J.; KÖNIG, W.; SCHÜTTE, R.; WENDT, O.; ZELEWSKI, S. (Hrsg.): Wirtschaftsinformatik und Wissenschaftstheorie. Wiesbaden 1997, S. 95-123.

MITTELMANN (1995)
MITTELMANN, R.: Intelligent unterstütztes systemtheoretisches Problemlösen. Dissertation, Universität Linz – Österreich. Linz 1995.

MIZOGUCHI/KITAMURA (2001)
MIZOGUCHI, R.; KITAMURA, Y.: Foundation of Knowledge Systematization: Role of Ontological Engineering. In: RAJKUMAR, R. (Hrsg.): Industrial Knowledge Management: A Micro-level Approach. Berlin et al. 2001, S. 17-36.

MONIN (2003)
MONIN, J. F.: Understanding Formal Methods. London et al. 2003.

MORDAU (1998)
MORDAU, J.: Die Integration formaler Methoden zur Spezifikation von Informationssystemen. Hamburg 1998.

MOTTA ET AL. (2000)
MOTTA, E.; SHUM, S.B.; DOMINGUE, J.: Ontology-driven document enrichment: principles, tools and applications. In: International Journal of Human-Computer Studies, 52. Jg. (2000), Nr. 6, S. 1071-1109.

MULHOLLAND ET AL. (2001)
MULHOLLAND, P.; ZDRAHAL, Z.; DOMINGUE, J.; HATALA, M.: A methodological approach to supporting organizational learning. In: International Journal of Human-Computer Studies, 55. Jg. (2001), Nr. 3, S. 337-367.

MÜLLER (1999)
MÜLLER, M.: Eine Methode zur automatischen Herleitung oder Widerleung einfacher Programmeigenschaften durch Generierung neuer Termersetzungsregeln. Dissertation, Universität Marburg. Marburg 1999.

MUSCHOLL (2001)
MUSCHOLL, K.M.: Interaktion und Koordination in Multiagentensystemen. Dissertation, Universität München. München 2001.

MYLOPOULOS ET AL. (1997)
MYLOPOULOS, J.; BORGIDA, A.; YU, E.: Representing Software Engineering Knowledge. In: Automated Software Engineering, 4. Jg. (1997), Nr. 3, S. 291-317.

NARAYANAN/MCILRAITH (2002)
NARAYANAN, S.; MCILRAITH, S.A.: Simulation, verification and automated composition of web services. In: O. HRSG.: WWW 2002. Proceedings of the Eleventh International World Wide Web Conference, WWW2002, Honolulu, Hawaii, USA, May 7-11 2002. New York 2002, S. 77-88.

NEUWEG (2001)
NEUWEG, H.G.: Könnerschaft und implizites Wissen. 2. Aufl., Münster et al. 2001.

NOY ET AL. (2001)
NOY, N.F.; SINTEK, M.; DECKER, S.; CRUBÉZY, M.; FERGERSON, R.W.; MUSEN, M.: Creating Semantic Web Contents with Protégé-2000. In: IEEE Intelligent Systems, 16. Jg. (2001), Nr. 2, S. 60-71.

NÜTTGENS/RUMP (2002)
NÜTTGENS, M.; RUMP, J.F.: Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK). In: DESEL, J.; WESKE, M. (Hrsg.): Promise 2002 – Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen. Berlin et al. 2002, S. 64-77.

O´LEARY (1997)
O´LEARY, D.E.: ImpredIments in the use of explicit ontologies for KBS development. In: International Journal of Human-Computer Studies, 46. Jg. (1997), Nr. 5-6, S. 327-337.

O´LEARY (1998)
O´LEARY, D.E.: Using AI in Knowledge Management: Knowledge Bases and Ontologies. In: IEEE Intelligent Systems, 13. Jg. (1998), Nr. 3, S. 34-39.

OBERSCHELP (1962)
OBERSCHELP, A.: Untersuchungen zur mehrsortigen Quantorenlogik. In: Mathematische Annalen, 145. Jg. (1962), S. 297-333.

OBERWEIS (1990)
OBERWEIS, A.: Zeitstrukturen in Informationssystemen. Dissertation, Universität Mannheim. Mannheim 1990.

OBERWEIS (1996)
OBERWEIS, A.: Modellierung und Ausführung von Workflows mit Petri-Netzen. Stuttgart – Leipzig 1996.

OBERWEIS/SANDER (1996)
OBERWEIS, A.; SANDER, P.: Information System Behavior Specification by High-Level Petri Nets. In: ACM Transactions on Information Systems, 14. Jg. (1996), Nr. 4, S. 380-420.

OGDEN/RICHARDS (1972)
OGDEN, C.K.; RICHARDS, I.A.: The Meaning of Meaning. 10. Aufl., London 1972.

OMELAYENKO (2002)
OMELAYENKO, B.: Ontology-Mediated Business Integration. In: GÓMEZ-PÉREZ, A.; BENJAMINS, R. (Hrsg.): Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 13th International Conference, EKAW 2002, Siguenza, Spain, October 1-4, 2002, Proceedings. Berlin et al. 2002, S. 264-269.

ONTOPRISE (2003)
O.V.: How to Use Ontobroker. Users and Developers Guide for the Ontobroker System Version 3.6. Karlsruhe 2003.

ORTNER (1997)
ORTNER, E.: Methodenneutraler Fachentwurf. Stuttgart – Leipzig 1997.

OUZOUNIS (2001)
OUZOUNIS, E.K.: An Agent-Based Platform for the Management of Dynamic Virtual Enterprises. Dissertation, Technische Universität Berlin. Berlin 2001.

OUZZANI ET AL. (2000)
OUZZANI, M.; BENTALLAH, B.; BOUGUETTAYA, A.: Ontological Approach for Information Discovery in Internet Databases. In: Distributed and Parallel Databases, 8. Jg. (2000), Nr. 3, S. 367-392.

PAHL (1996)
PAHL, C.: Grundlagen für die formale Spezifikation modularer zustandsbasierter Systeme. Dissertation, Universität Dortmund. Dortmund 1996.

PAPAZOGLOU (2002)
PAPAZOGLOU, M.P.: The World of e-Business: Web-Services, Workflows, and Business Transactions. In: BUSSLER, C.; HULL, R.; MCILRAITH, S.A.; ORLOWSKA, M.E.; PERNICI, B.; YANG, J. (Hrsg.): Web Services, E-Business, and the Semantic Web, CAiSE 2002 International Workshop, WES 2002, Toronto, Canada, May 27-28, 2002, Revised Papers. Berlin et al. 2002, S. 153-173.

PAPAZOGLOU (2003)
PAPAZOGLOU, M.P.: Web Services and Business Transactions. In: World Wide Web, 6. Jg. (2003), Nr. 1, S. 49-91.

PATIG (2001)
PATIG, S.: Flexible Produktionsfeinplanung mit Hilfe von Planungsschritten. Dissertation, Universität Magdeburg. Magdeburg 2001.

PATIG (2001A)
PATIG, S.: Überlegungen zur theoretischen Fundierung der Disziplin Wirtschaftsinformatik, ausgehend von der allgemeinen Systemtheorie. In: Journal for General Philosophy of Science, 32. Jg. (2001), S. 39-64.

PATIG (2004)
PATIG, S.: Zur Ausdrucksstärke der Stammdaten des Advanced Planning und Scheduling. In: Wirtschaftsinformatik, 46. Jg. (2004), Nr. 2, S. 97-106.

PETKOFF (1999)
PETKOFF, B.: Die Kybernetik II. Ordnung – eine methodologische Basis der Wirtschaftsinformatik? In: BECKER, J.; KÖNIG, W.; SCHÜTTE, R.; WENDT, O.; ZELEWSKI, S. (Hrsg.): Wirtschaftsinformatik und Wissenschaftstheorie. Wiesbaden 1997, S. 242-287.

PFEIFER (2000)
PFEIFER, F.-P.: Vokabularunterstützung in der Unternehmensmodellierung durch Thesauri. In: SCHMIDT, H. (Hrsg.): Modellierung betrieblicher Informationssysteme, Proceedings der MobIS Fachtagung 2000, Rundbrief der GI-Fachgruppe 5.10, 7. Jg., Nr. 1, Siegen 2000, S. 229-250.

PFLÜGLMAYER (2001)
PFLÜGLMAYER, M.: Informations- und Kommunikationstechnologien zur Qualitätsverbesserung im Krankenhaus. Dissertation, Universität Linz. Linz 2001.

PHILIPPI (1999)
PHILIPPI, S.: Synthese von Petri-Netzen und objektorientierten Konzepten. Koblenz 1999.

PINKER (1996)
PINKER, S.: Der Sprachinstinkt. München 1996.

PIRLEIN/STUDER (1995)
PIRLEIN, T.; STUDER, R.: An environment for reusing ontologies within a knowledge engineering approach. In: International Journal of Human-Computer Studies, 43. Jg. (1995), Nr. 5-6, S. 945-965.

POLANYI (1985)
POLANYI, M.: Implizites Wissen. Frankfurt a.M. 1985.

POPPER (1994)
POPPER, K.: Logik der Forschung. 10. Aufl., Tübingen 1994.

PRASSE (2002)
PRASSE, M.: Entwicklung und Formalisierung eines objektorientierten Sprachmodells als Grundlage für MEMO-OML. Dissertation, Universität Koblenz-Landau. Koblenz 2002.

PREECE ET AL. (2001)
PREECE, A.; HUI, K.; GRAY, A.; MARTI, P.; BENCH-CAPON, T.; CUI, Z.; JONES, D.: KRAFT: An Agent Architecture for Knowledge Fusion. In: International Journal of Cooperative Information Systems, 10. Jg. (2001), Nr. 1/2, S. 197-224.

QUINE (1969)
QUINE, W.v.O.: Grundzüge der Logik. Frankfurt a.M. 1969.

RADA ET AL. (1990)
RADA, R.; DUNNE, P.E.S.; BARLOW, J.: EXPERTEXT: From Semantic Nets to Logic Petri Nets. In: Expert Systems With Applications, 1. Jg. (1990), Nr. 1, S. 51-62.

RAM/KHATRI (2005)
RAM, S.; KHATRI, V.: A comprehensive framework for modeling set-based business rules during concetpual database design. In: Information Systems, 30. Jg. (2005), Nr. 2, S. 89-118.

RAPAPORT (2003)
RAPAPORT, W.J.: What Did You Mean by That? Misunderstanding, Negotiation, and Syntactic Semantics. In: Minds and Machines, 13. Jg. (2003), Nr. 3, S. 397-427.

RAUH/STICKEL (1997)
RAUH, O.; STICKEL, E.: Konzeptuelle Datenmodellierung. Stuttgart 1997.

REISIG (1991)
REISIG, W.: Petrinetze. 2. Aufl., Berlin et al. 1991.

REISIG (1991A)
REISIG, W.: Petri Nets and Algebraic Specifications. In: Theoretical Computer Science, 80. Jg. (1991), Nr. 1, S. 1-34.

RÖBBECKE (1995)
RÖBBECKE, M.: Erweiterung von Petri-Netzen für den Einsatz in der Wissensverarbeitung. Dissertation, Universität Clausthal. Clausthal 1995.

ROPOHL (1979)
ROPOHL, G.: Eine Systemtheorie der Technik. München – Wien 1979.

ROSCA ET AL. (2002)
ROSCA, D.; GREENSPAN, S.; WILD, C.: Enterprise Modeling and Decision-Support for Automating the Business Rules Lifecycle. In: Automated Software Engineering, 9. Jg. (2002), Nr. 4, S. 361-404.

ROSENKRANZ (2002)
ROSENKRANZ, F.: Geschäftsprozesse – Modell- und computergestützte Planung. Berlin et al. 2002.

ROSENSTENGEL/WINAND (1991)
ROSENSTENGEL, B.; WINAND, U.: Petri-Netze. 4. Aufl., Braunschweig 1991.

ROTHMALER (1995)
ROTHMALER, P.: Einführung in die Modelltheorie. Heidelberg et al. 1995.

ROZENBERG/ENGELFRIET (1998)
ROZENBERG, G.; ENGELFRIET, J.: Elementary Net Systems. In: REISIG, W.; ROZENBERG, G. (Hrsg.): Lectures on Petri Nets I: Basic Models. Berlin et al. 1998, S. 12-121.

SAAKE (1993)
SAAKE, G.: Objektorientierte Spezifikation von Informationssystemen. Stuttgart – Leipzig 1993.

SCANNAPIECO ET AL. (2004)
SCANNAPIECO, M.; VIRGILLITO, A.; MARCHETTI, C.; MECELLA, M.; BALDONI, R.: The DaQuinCIS architecture: a platform for exchanging and improving data quality in cooperative information systems. In: Information Systems, 29. Jg. (2004), Nr. 7, S. 551-582.

SCHEFE (1999)
SCHEFE, P.: Softwaretechnik und Erkenntnistheorie. In: Informatik Spektrum, 22. Jg. (1999), Nr. 2, S. 122-135.

SCHMID (1998)
SCHMID, C.: Beitrag zur Modellierung und Analyse dynamischer Unternehmensmodelle. Dissertation, Universität Karlsruhe. Aachen 1998.

SCHMID/KINDSMÜLLER (1996)
SCHMID, U.; KINDSMÜLLER, M.C.: Kognitive Modellierung. Heidelberg et al. 1996.

SCHMIDT-SCHAUß (1989)
SCHMIDT-SCHAUß, M.: Computational Aspects of an Order-Sorted Logic with Term Declarations. Berlin et al. 1989.

SCHÖNING (1992)
SCHÖNING, U.: Logik für Informatiker. 3. Aufl. Mannheim et al. 1992.

SCHREYÖGG (2003)
SCHREYÖGG, G.: Organisation. 4. Aufl., Wiesbaden 2003.

SCHULZE (2001)
SCHULZE, D.: Grundlagen der wissensbasierten Konstruktion von Modellen betrieblicher Systeme. Aachen 2001.

SCHÜTTE (1998)
SCHÜTTE, R.: Grundsätze ordnungsmäßiger Referenzmodellierung. Dissertation, Universität Münster. Wiesbaden 1998.

SCHÜTTE (1999)
SCHÜTTE, R.: Basispositionen in der Wirtschaftsinformatik – ein gemäßigt-konstruktivistisches Programm. In: BECKER, J.; KÖNIG, W.; SCHÜTTE, R.; WENDT, O.; ZELEWSKI, S. (Hrsg.): Wirtschaftsinformatik und Wissenschaftstheorie. Wiesbaden 1997, S. 211-241.

SCHWEGMANN (1999)
SCHWEGMANN, A.: Objektorientierte Referenzmodellierung. Wiesbaden 1999.

SEIBT (2001)
SEIBT, J.: Formal process ontology. In: WELTY, C.; SMITH, B. (Hrsg.): Formal Ontology in Information Systems. 2nd International Conference on Formal Ontology in Information Systems, FOIS 2001, Ogunquit, Maine, USA, October 17-19, 2001, Proceedings. New York 2001, S. 333-345.

SMITH (1996)
SMITH, B.: A Theory of Parts and Boundaries. In: Data & Knowledge Engineering, 20. Jg. (1996), Nr. 3, S. 287-303.

SMITH (1998)
SMITH, E.: Principles of High-Level Net Theory. In: REISIG, W.; ROZENBERG, G. (Hrsg.): Lectures on Petri Nets I: Basic Models. Berlin et al. 1998, S. 174-210.

SMOLKA (1989)
SMOLKA, G.: Logic Programming over Polymorphically Order-Sorted Types. Dissertation, Universität Kaiserslautern. Kaiserslautern 1989.

SOWA (1984)
SOWA, J.: Conceptual Structures. Reading et al. 1984.

SOWA (1995)
SOWA, J.: Top-level ontological categories. In: International Journal of Human-Computer Studies, 43. Jg. (1995), Nr. 5-6, S. 669-685.

SOWA (2000)
SOWA, J.: Knowledge Representation. Pacific Grove et al. 2000.

SOWA (2000A)
SOWA, J.: Ontology, Metadata and Semiotics. In: GANTER, B.; MINEAU, G.W. (Hrsg.): Conceptual Structures: Logical, Linguistic, and Computational Issues, 8th International Conference on Conceptual Structures, ICCS 2000, Darmstadt, Germany, August 14-18, 2000, Proceedings. Berlin et al. 2000. S. 55-81.

SPECK (2001)
SPECK, M.C.: Geschäftsprozessorientierte Datenmodellierung. Dissertation, Universität Münster. Berlin 2001.

STACHOWIAK (1973)
STACHOWIAK, H.: Allgemeine Modelltheorie. Wien – New York 1973.

STACHOWIAK (1983)
STACHOWIAK, H.: Erkenntnisstufen zum Systematischen Neopragmatismus und zur Allgemeinen Modelltheorie. In: STACHOWIAK, H. (Hrsg.): Modelle – Konstruktion der Wirklichkeit. München 1983, S. 87-146.

STARKE (1990)
STARKE, P.H.: Analyse von Petri-Netz-Modellen. Stuttgart et al. 1990.

STEELS (1998)
STEELS, L.: The Origins of Ontologies and Communication Conventions in Multi-Agent Systems. In: Autonomous Agents and Multi-Agent Systems, 1. Jg. (1998), Nr. 2, S. 169-194.

STEHR ET AL. (2001)
STEHR, M-O.; MESEGUER, J.; ÖLVECZKY, P. C.: Rewriting Logic as a Unifying Framework for Petri Nets. Aus: EHRIG, H.; JUHÀS, G.; PADBERG, J.; ROZENBERG, G. (Hrsg.): Unifying Petri Nets, Advances in Petri Nets. Berlin et al. 2001. S. 250-303.

STEIMANN (2000)
STEIMANN, F.: Formale Modellierung mit Rollen. Habilitationsschrift an der Universität Hannover. Hannover 2000.

STEVENS ET AL. (2004)
STEVENS, R.; WROE, C.; LORD, P.; GOBLE, C.: Ontologies in Bioinformatics. In: STAAB, S.; STUDER, R. (Hrsg.): Handbook on Ontologies. Berlin et al. 2004, S. 635-657.

STOREY ET AL. (1998)
STOREY, V.C.; DEY, D.; ULLRICH, H.; SUNDARESAN, S.: An Ontology-Based Expert System for Database Design. In: Data & Knowledge Engineering, 28. Jg. (1998), Nr. 1, S. 31-46.

STUCKENSCHMIDT (2003)
STUCKENSCHMIDT, H.: Ontology-Based Informtion Sharing in Weakly Structured Environments. Dissertation, Universität Amsterdam – Niederlande. Amsterdam 2003.

STUDER ET AL. (1998)
STUDER, R.; BENJAMINS, V.R.; FENSEL, D.: Knowledge Engineering: Principles and methods. In: Data & Knowledge Engineering, 25. Jg. (1998), Nr. 1-2, S. 161-197.

SUGUMARAN/STOREY (2002)
SUGUMARAN, V.; STOREY, V.C.: Ontologies for conceptual modeling: their creation, use, and management. In: Data & Knowledge Engineering, 42. Jg. (2002), Nr. 3, S. 251-271.

SURE (2003)
SURE, Y.: Metodology, Tools & Case Studies for Ontology based Knowledge Management. Dissertation, Universität Karlsruhe. Karlsruhe 2003.

SYDOW (1992)
SYDOW, J.: Strategische Netzwerke. Wiesbaden 1992.

TACKEN (2001)
TACKEN, J.: Eine Pr/T-Netz basierte, durchgängige Entwurfsmethodik für eingebettete Realzeitsysteme. Dissertation, Universität Paderborn. Aachen 2001.

TAMMA/BENCH-CAPON (2002)
TAMMA, V.A.M.; BENCH-CAPON, T.: Attribute Meta-properties for Formal Ontological Analysis. In: GÓMEZ-PÉREZ, A.; BENJAMINS, R. (Hrsg.): Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 13th International Conference, EKAW 2002, Siguenza, Spain, October 1-4, 2002, Proceedings. Berlin et al. 2002, S. 301-316.

ULLRICH ET AL. (2000)
ULLRICH, H.; PURAO, S.; STOREY, V.C.: An Ontology for Classifying the Semantics of Relationships in Database Design. In: BOUZEGHOUB, M.; KEDAD, Z.; MÉTAIS, E. (Hrsg.): Natural Language Processing and Information Systems, 5th International Conference on Applications of Natural Language to Information Systems, NLDB 2000, Versailles, France, June 28-30, 2000, Revised Papers. Berlin et al. 2000, S. 91-102.

UPHOFF (1997)
UPHOFF, H.: Ein Ansatz zur Integration von Pfadausdrücken in Frame-Logik. Dissertation, Universität Freiburg im Breisgau. Freiburg im Breisgau 1997.

USCHOLD (2002)
USCHOLD, M.: A semantic continuum on the semantic web. In: The Knowledge Engineering Review, 17. Jg. (2002), Nr. 1, S. 87-91.

USCHOLD ET AL. (1998)
USCHOLD, M.; KING, M.; MORALEE, S.; ZORGIOS, Y.: The Enterprise Ontology. In: The Knowledge Engineering Review, 13. Jg. (1998), Nr. 1, S. 31-89.

USCHOLD/GRUNINGER (1996)
USCHOLD, M.; GRUNINGER, M.: Ontologies: principles, methods and applications. In: The Knowledge Engineering Review, 11. Jg. (1996), Nr. 2, S. 93-136.

VAN DER AALST/VAN HEE (2002)
VAN DER AALST, W.; VAN HEE, K.: Workflow Management. Cambridge – London 2002.

VAN ECK ET AL. (1998)
VAN ECK, P.; ENGELFRIET, J.; FENSEL, D.; VAN HARMELEN, F.; VENEMA, Y.; WILLEMS, M.: Specification of Dynamics for Knowledge-Based Systems. In: FREITAG , B.; DECKER, H.; KIFER, M.; VORONKOV, A. (Hrsg.): Transactions and Change in Logic Databases. Berlin et al. 1998, S. 37-68.

VAN EMDEN/KOWALSKI (1976)
VAN EMDEN, M.H.; KOWALSKI, R.A.: The Semantics of Predicate Logic as a Programming Language. In: Journal of the ACM, 23. Jg. (1976), Nr. 4, S. 733-742.

VAN HEIJST ET AL. (1997)
VAN HEIJST, G.; SCHREIBER, A.T.; WIELINGA, B.J.: Using explicit ontologies in KBS development. In: International Journal of Human-Computer Studies, 45. Jg. (1997), Nr. 2-3, S. 183-292.

VAN HOREBEEK/LEWI (1989)
VAN HOREBEEK, I.; LEWI, J.: Algebraic Specifications in Software Engineering. Berlin et al. 1989.

VERHAREN/DIGNUM (1997)
VERHAREN, E.; DIGNUM, F.: Cooperative Information Agents and Communication. In: KANDZIA, P.; KLUSCH, M. (Hrsg.): Cooperative Information Agents, First International Workshop, CIA' 97, Kiel, Germany, February 26-28, 1997, Proceedings. Berlin et al. 1997, S. 195-209.

VISSER ET AL. (1999)
VISSER, P.R.S.; BEER, M.D.; BENCH-CAPON, T.J.M.; DIAZ, B.M.; SHAVE, M.J.R.: Resolving Ontological Heterogeneity in the KRAFT Project. In: BENCH-CAPON, T.; SODA, G.; MIN TJOA, A. (Hrsg.): Database and Expert Systems Applications. 10th International Conference, DEXA '99, Florence, Italy, August 30 – September 3, 1999, Proceedings. Berlin et al. 1999, S. 668-677.

VISSER/BENCH-CAPON (1998)
VISSER, P.R.S.; BENCH-CAPON, T.: A Comparison of Four Ontologies for the Design of Legal Knowledge Systems. In: Artificial Intelligence and Law, 6. Jg. (1998), Nr. 1, S. 27-57.

VON DER OELSNITZ (2003)
VON DER OELSNITZ, D.: Kooperation: Entwicklung und Verknüpfung von Kernkompetenzen. In: ZENTES, J.; SWOBODA, B.; MORSCHETT, D. (Hrsg.): Kooperationen, Allianzen und Netzwerke. Wiesbaden 2003. S. 185-210.

VON KUTSCHERA (1967)
VON KUTSCHERA, F.: Elementare Logik. Wien 1967.

VOSSEN (1994)
VOSSEN, G.: Datenmodelle, Datenbanksprachen und Datenbank-Management-Systeme. 2. Aufl., Bonn et al. 1994.

WEBER (2002)
WEBER, M.: Allgemeine Konzepte zur software-technischen Unterstützung verschiedener Petrinetz-Typen. Dissertation, Technische Universität Berlin. Berlin 2002.

WEDEKIND ET AL. (2004)
WEDEKIND, H.; ORTNER, E.; INHETVEEN, R.: Informatik als Grundbildung – Teil III: Gleichheit und Abstarktion. In: Informatik Spektrum, 27. Jg. (2004), H. 4, S. 337-342.

WEDEKIND ET AL. (2004A)
WEDEKIND, H.; ORTNER, E.; INHETVEN, R.: Informatik als Grundbildung – Teil IV: Objektsprache/Metasprache. In: Informatik Spektrum, 27. Jg. (2004), H. 5, S. 459-466.

WEISSMAHR (1991)
WEISSMAHR, B.: Ontologie. 2. Aufl., Stuttgart et al. 1991.

WEITZ (2000)
WEITZ, W.: Integrierte Dokumenten- und Ablaufmodellierung im Electronic Commerce. Dissertation, Universität Karlsruhe. Aachen 2000.

WELTY/FERRUCCI (1999)
WELTY, C.; FERRUCCI, D.A.: Instances and Classes in Software Engineering. In: Intelligence, 10. Jg. (1999), Nr. 2, S. 18-23.

WELTY/GUARINO (2001)
WELTY, C.; GUARINO, N.: Supporting ontological analysis of taxonomic relationships. In: Data & Knowledge Engineering, 39. Jg. (2001), Nr. 1, S. 51-74.

WESTERMANN (2000)
WESTERMANN, R.: Wissenschaftstheorie und Experimentalmethodik. Göttingen et al. 2000.

WHORF (2003)
WHORF, B.L.: Sprache – Denken – Wirklichkeit. 24. Aufl., Reinbek 2003.

WIENBERG (2001)
WIENBERG, F.: Informations- und prozeßorientierte Modellierung verteilter Systeme auf der Basis von Feature-Structure-Netzen. Dissertation, Universität Hamburg. Hamburg 2001.

WIMMER/WIMMER (1992)
WIMMER, K.; WIMMER, N.: Conceptual modelling based on ontological principles. In: Knowledge Acquisition, 4. Jg. (1992), S. 387-406.

WINOGRAD/FLORES (1999)
WINOGRAD, T.; FLORES, F.: Understanding Computers and Cognition. 14. Aufl., Norwood 1999.

WOLF (2001)
WOLF, S.: Wissenschaftstheoretische und fachmethodische Grundlagen der Konstruktion von generischen Referenzmodellen betrieblicher Systeme. Dissertation, Universität Bamberg. Aachen 2001.

WROE ET AL. (2003)
WROE, C.; STEVENS, R.; GOBLE, C.; ROBERTS, A.; GREENWOOD, M.: A Suite of Daml+Oil Ontologies to Describe Bioinformatics Web Services and Data. In: International Journal of Cooperative Information Systems, 12. Jg. (2003), Nr. 2, S. 197-224.

WÜTHRICH/PHILIPP (1998)
WÜTHRICH, H.A.; PHILIPP, A.: Grenzenlose Chancen durch Virtualisierung!? In: Zeitschrift Führung & Organisat, 69. Jg. (1998), Nr. 4, S. 201-206.

XU ET AL. (2002)
XU, D.; VOLZ, R.A.; IOERGER, T.R.; YEN, J.: Modeling and verifying multi-agent behaviors using predicate/transition nets. In: O. HRSG.: Proceedings of the 14th international conference on Software engineering and knowledge engineering, July 15-19, 2002, Ischia, Italy. New York 2002, S. 193-200.

ZELEWSKI (1995)
ZELEWSKI, S.: Petrinetzbasierte Modellierung komplexer Produktionssysteme (Projekt PEMOPS), Bd. 1-10. Arbeitsberichte des Instituts für Produktionswirtschaft und Industrielle Informationswirtschaft (Nr. 5-15), Universität Leipzig. Leipzig 1995.

ZELEWSKI (1996)
ZELEWSKI, S.: Eignung von Petri-Netzen für die Modellierung komplexer Realsysteme – Beurteilungskriterien. In: Wirtschaftsinformatik, 38. Jg. (1996), Nr. 4, S. 369-381.

ZELEWSKI (2005)
ZELEWSKI, S.: Einführung in das Themenfeld "Ontologien" aus informations- und betriebswirtschaftlicher Perspektive. In: ZELEWSKI, S.; ALAN, Y.; ALPARSLAN, A.; DITTMANN, L.; WEICHELT, T. (Hrsg.): Ontologiebasierte Kompetenzmanagementsysteme – Grundlagen, Konzepte, Anwendungen. [Im Erscheinen] Berlin 2005.

ZIMMER (2001)
ZIMMER, T.: Petri-Netz-Konzepte für die Simulation verteilter betrieblicher Abläufe. Dissertation, Universität Franfurt a. M. Aachen 2001.