

AUS DER ABTEILUNG  
E-HEALTH COMPETENCE CENTER  
PRIV.-DOZ. DR. RER. NAT. HABIL. BERND BLOBEL  
DER FAKULTÄT FÜR MEDIZIN  
DER UNIVERSITÄT REGENSBURG

ENTWICKLUNG EINER ONTOLOGIEBASIERTEN ARCHITEKTUR ZUR  
SICHERUNG SEMANTISCHER INTEROPERABILITÄT ZWISCHEN  
KOMMUNIKATIONSSTANDARDS IM GESUNDHEITSWESEN

Inaugural – Dissertation  
zur Erlangung des Doktorgrades  
der Biomedizinischen Wissenschaften

der  
Fakultät für Medizin  
der Universität Regensburg

vorgelegt von  
Frank Oemig

2011

Dekan: Prof. Dr. Bernhard Weber  
1. Berichterstatter: PD Dr. Bernd Blobel  
2. Berichterstatter: Prof. Dr. Michael Nerlich  
Tag der mündlichen Prüfung: 22.Dez. 2010

## **Eidesstattliche Versicherung**

Ich, Frank Oemig geboren am 11.7.1963 in Dortmund erkläre hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe.

Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Insbesondere habe ich nicht die entgeltliche Hilfe von Vermittlungs- bzw. Beratungsdiensten (Promotionsberater oder andere Personen) in Anspruch genommen.

Niemand hat von mir unmittelbar oder mittelbar Geld oder geldwerte Leistungen für Arbeit erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.

Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.

---

Mülheim, den 10.02.2011



## **Danksagung**

An dieser Stelle möchte ich all denjenigen ein herzliches Dankeschön aussprechen, die mir mit unterschiedlichster Unterstützung ermöglicht haben, diese Arbeit zu erstellen.

Zu ganz besonderem Dank bin ich dabei Herrn PD Dr. Bernd Blobel verpflichtet, der mich unermüdlich mit Hinweisen und konstruktiver Kritik durch die Jahre hindurch begleitet hat. Dazu gehören eine hervorragende Betreuung mit zahlreichen Gesprächen inkl. privater „Lectures“ sowie vielen E-Mails und die stets umgehenden und ausführlichen Reaktionen, die dann wieder zu neuen Diskussionen geführt haben.

Herrn Prof. Dr. med. Michael Nerlich danke ich für sein Interesse und vor allem für seinen Einsatz, mit dem er mir die Möglichkeit, an der Universität Regensburg als externer Doktorand angenommen zu werden und diese Dissertation zu schreiben, erst eröffnet hat.

Auch Apl. Prof. Dr. med. Stefan Schulz möchte ich an dieser Stelle für die vielen Diskussionen bedanken, die für mich eine große Hilfe bei der Erstellung dieser Arbeit waren.

An dieser Stelle darf ich auch meine Familie nicht vergessen, die mich in der Beständigkeit meines Engagements bestärkt hat.

Vor allem aber möchte ich meiner Frau Anja für ihre endlose Geduld, ihre zahlreiche Ermutigungen und ihre einfach großartige Unterstützung danken. Ohne diese Rückendeckung wäre diese Arbeit nicht möglich gewesen.



<b>Inhaltsverzeichnis</b>	<b>Seite</b>
<b>Inhaltsverzeichnis.....</b>	<b>1</b>
<b>A. Motivation.....</b>	<b>7</b>
1. Einleitung.....	7
1.1. Paradigmenwechsel im Gesundheitswesen.....	7
1.2. Kommunikation und Kooperation im Gesundheitswesen.....	8
1.3. Integrierte Versorgung und interoperable Systeme.....	8
1.4. Motivation zur Kooperation.....	9
1.5. Informationszyklus.....	10
2. Aufgabenstellung.....	11
3. Gliederung der Arbeit.....	12
<b>B. Grundlagen.....</b>	<b>14</b>
4. Hintergrundinformationen.....	14
4.1. Anforderungen an interoperable Systeme.....	14
4.2. Gegenüberstellung der Standards.....	16
4.3. Ablauf der Kommunikation.....	17
5. Definitionen.....	21
5.1. Modell.....	21
5.2. Architektur.....	21
5.3. Interoperabilität.....	21
5.4. Linguistik.....	22
5.4.1. Syntax.....	23
5.4.2. Semantik.....	23
5.4.3. Semiotisches Dreieck.....	24
5.4.4. Pragmatismus.....	25
5.4.5. Pragmatik.....	25
5.5. Ontologie.....	25
5.6. Epistemologie (Erkenntnistheorie).....	26
5.7. Wissen.....	27
5.8. Wissensrepräsentation.....	28
5.9. Wissensbasis.....	28
5.10. Konzept.....	28
5.11. Beziehungen (Relationen).....	29
5.12. Klassifikation.....	30
<b>C. Methoden.....</b>	<b>32</b>
6. Rahmenarchitektur.....	32
6.1. RM-ODP –Reference Model for Open Distributed Processing.....	32
6.2. SAEAF – Service Aware Enterprise Architecture Framework.....	34
6.3. Das generische Komponentenmodell.....	35

<b>Inhaltsverzeichnis</b>	<b>Seite</b>
6.3.1. Basisprinzipien.....	38
6.3.2. Vorteile.....	38
6.4. Einschränkung für diese Arbeit .....	39
<b>7. Wissen und Wissensrepräsentation.....</b>	<b>41</b>
7.1. Ontologie.....	42
7.1.1. Einsatz von Agenten .....	43
7.1.2. Hierarchien/Typisierung .....	44
7.1.3. Repräsentationsformen .....	46
7.1.4. T-Box/A-Box.....	46
7.1.5. Nutzung des Wissens.....	46
7.1.6. First-Order-Logik Definition .....	47
7.1.7. Reasoning .....	48
7.2. Ontologie- und KR-Sprachen .....	48
7.2.1. Prolog und LISP .....	48
7.2.2. Resource Description Framework (RDF) .....	48
7.2.3. Web Ontology Language (OWL) .....	49
<b>8. Ordnungsstrukturen für Wissensrepräsentation.....</b>	<b>49</b>
8.1. Beobachtungseinheiten .....	50
8.2. Begriffsmengen .....	50
8.2.1. Ordnungsstrukturen .....	50
8.2.2. Vokabular.....	51
8.2.3. Thesaurus .....	52
8.2.4. Terminologie .....	52
8.2.5. Nomenklatur.....	52
8.2.6. Taxonomie .....	53
8.2.7. Partonomie.....	53
8.2.8. Regeln .....	53
8.2.9. Frames .....	53
8.3. UML-Klassendiagramm der Begriffsmengen.....	53
8.4. Einordnung in das GCM .....	55
<b>D. Ergebnisse .....</b>	<b>58</b>
<b>9. Analyse der Kommunikationsstandards .....</b>	<b>58</b>
9.1. UML-Modell für HL7 v2.x.....	59
9.1.1. Anmerkungen zum HL7-v2.x-Modell .....	63
9.1.2. Anmerkungen zum HL7-v2.x-Datentypen-Modell.....	68
9.2. UML-Modell für HL7 V3 .....	68
<b>10. Ontologien für Kommunikationsstandards.....</b>	<b>69</b>
10.1. Grundlagen der Repräsentation medizinischen Wissens.....	71
10.1.1. Relationales Repräsentationsformat von Snomed CT.....	71
10.1.2. Relationen und Achsen .....	72
10.1.3. Konvertierung nach OWL.....	73
10.2. Ontologie-Struktur .....	73
10.2.1. Überlegungen .....	73
10.2.2. Information Artefact Ontology .....	75
10.2.3. Basis-Ontologie (Top-Level Ontology) .....	77
10.2.4. RO : Relationship Ontology .....	78
10.3. Diskussion konkurrierender Ansätze für eine Repräsentation von HL7-Kommunikationsstandards.....	79



<b>Inhaltsverzeichnis</b>	<b>Seite</b>
10.4. Die Communication Standards Ontology als Referenz-Ontologie.....	80
10.4.1. Top-Level-Ontologie mit nächster Ebene .....	81
10.4.2. CSO + BFO: Hierarchie .....	81
10.4.3. Eigenschaften der Informationsobjekte (als Qualitäten).....	83
10.4.4. Optionalität .....	84
10.4.5. Beziehungen/Relationen (Object-Properties) .....	86
10.4.6. Data-Properties .....	87
10.4.7. Instanzen .....	87
10.5. Abbildung HL7 V3 $\Rightarrow$ OWL .....	87
10.5.1. Datenquelle .....	87
10.5.2. Einordnung in das GCM .....	88
10.5.3. Darstellung der Informationen .....	90
10.5.4. OWL-Elemente für HL7 V3 .....	90
10.5.5. Fehler in den Quelldaten.....	96
10.6. Abbildung HL7 v2.x $\Rightarrow$ OWL.....	97
10.6.1. Datenquelle .....	97
10.6.2. Einordnung in das GCM .....	97
10.6.3. OWL-Elemente für HL7 v2.x.....	98
10.6.4. Beispiel-Instanz PID-Segment.....	105
10.7. Zwischenfazit .....	108
10.8. Konsistenzregeln .....	108
10.8.1. Beispielregeln .....	108
10.8.2. Umsetzung mittels SWRL.....	109
<b>11. Analyse der Fachdomäne.....</b>	<b>109</b>
11.1. Beispiel: Name, Adresse und Telefonnummer .....	110
11.2. Beispiel: "Fall" .....	113
11.3. Beispiel: Diagnosen.....	114
11.4. Beispiel: Scores und Assessments.....	115
11.5. Einsatz der BFO (Basic Formal Ontology) und ACGT (Advancing Clinico Genomic Trials on Cancer).....	118
<b>12. Ontologien für die Fachdomäne.....</b>	<b>124</b>
12.1. Beispiel Name/Adresse .....	124
12.2. Beispiel: Fall.....	124
12.3. Beispiel: Scores .....	125
12.3.1. UML-OWL-Modell .....	125
12.3.2. Integritätsregeln für das OWL-Modell .....	128
12.3.3. Weitere Integration in ACGT .....	129
<b>13. Mapping .....</b>	<b>130</b>
13.1. Basiskonzepte .....	130
13.1.1. Beispiel-Mapping anhand des Patientennamens.....	130
13.1.2. Beispiel-Mapping anhand der Telefonnummer .....	132
13.2. Darstellung in Beschreibungslogik .....	132
13.3. Ablauf des Mappings .....	134
13.4. Ergänzung zu CSO .....	134
13.4.1. Mapping .....	135
13.4.2. RegExpression .....	136
13.4.3. Data-Properties.....	140
13.4.4. Mapping für v2.x .....	140
13.4.5. Mapping für V3 .....	142
13.4.6. Übersichtsschaubild .....	143
<b>14. praktische Erzeugung der OWL-Dateien .....</b>	<b>145</b>

<b>Inhaltsverzeichnis</b>	<b>Seite</b>
14.1. Erzeugung der Ontologien.....	145
14.1.1. Ontologiegrundstruktur für die Kommunikationsstandards .....	147
14.1.2. Ontologiegrundstruktur für das Mapping.....	148
14.2. Repräsentation von Ontologien mittels OWL.....	149
14.2.1. Scripting .....	149
14.2.2. HL7 v2.x .....	152
14.2.3. HL7 V3.....	153
14.2.4. Import-Struktur und Namespaces.....	153
14.2.5. Namespaces und URLs zum Managen ununterscheidbarer Ontologie-Versionen .....	154
14.2.6. Klasse vs. Instanz .....	154
14.2.7. Reasoning .....	155
14.2.8. Protégé 3.4 vs. 4.0.2 vs. 4.1.....	156
14.2.9. Fehler in Protégé 3.4 .....	156
14.2.10. Fehler in Protégé 4.0.2.....	156
14.2.11. Festlegung eines konsistenten Subsets.....	157
14.2.12. Implementierungsaspekte.....	158
14.2.13. Eine kleine Statistik .....	158
15. Bereitstellung der Ergebnisse.....	160
15.1. Ontologien .....	160
15.2. Ausgelagerte Anhänge.....	162
<b>E. Diskussion.....</b>	<b>164</b>
16. Charakterisierung der entwickelten Lösung.....	164
17. Bewertung des Nutzens.....	165
17.1. Bewertung der eigenen Arbeit .....	165
17.2. Zukünftige Nutzung der Ergebnisse .....	167
17.3. Übertragung auf der Ergebnisse auf Persistenzstrukturen.....	167
17.3.1. UML-Modell für Business-Objekte .....	167
17.3.2. UML-Modell für relationale Datenbanken.....	168
17.3.3. Abbildungsprozess.....	169
18. offene Punkte .....	170
18.1. Ausblick auf zukünftige Arbeiten .....	171
18.2. Architekturskizze der Anwendung .....	171
19. Zusammenfassung und Schlussfolgerungen.....	172
<b>F. Anhänge.....</b>	<b>174</b>
20. Anhang A: Einführung in die Grundlagen.....	174
20.1. RDF: Resource Description Framework .....	174
20.2. OWL: Web Ontology Language .....	176
20.2.1. Die Untersprachen von OWL 1.....	178
20.2.2. Die Sprachkonstrukte.....	178
20.2.3. OWL 2.....	179
20.2.4. SWRL.....	180
20.3. OPPL: Ontology Preprocessor Language.....	180

---

<b>Inhaltsverzeichnis</b>	<b>Seite</b>
21. Anhang B: OWL-Spezifikationen .....	181
21.1. Umsetzung von HL7 V3 MIF nach OWL .....	181
21.1.1. RIM .....	181
21.1.2. Vocabulary .....	185
21.1.3. R-MIM: MIF2-Struktur.....	189
21.1.4. Datentypen.....	197
21.1.5. CMET .....	199
21.2. Umsetzung der verschiedenen HL7-v2-Strukturen nach OWL.....	200
22. Verzeichnisse.....	201
22.1. Abkürzungen.....	201
22.2. Abbildungen .....	203
22.3. Tabellen.....	206
22.4. Literatur .....	208
22.5. Index .....	226



## A. Motivation

### 1. Einleitung

#### 1.1. *Paradigmenwechsel im Gesundheitswesen*

Das Gesundheitswesen steht vor der Herausforderung, die Qualität und Sicherheit der Patientenversorgung sowie die Effizienz der Versorgungsprozesse zu steigern, und das trotz der komplizierter werdenden Rahmenbedingungen wie der höheren Servicenachfrage infolge der demografischen Entwicklung mit einer alternden Bevölkerung, den multiple und zunehmend chronisch Erkrankten; dem wachsenden Anspruchdenken, dem medizinischen und biomedizintechnischen Fortschritt, kombiniert mit steigenden Aufwendungen und dem Beitragsschwund in der gesetzlichen Krankenversicherung [Blob2002]. Die Antwort in den Industrie-, aber auch den meisten Entwicklungsländern ist ein Paradigmenwechsel hin zu einem zunehmend spezialisierten, arbeitsteiligen und dezentralisierten System, d.h. der Übergang von einer organisationszentrierten zu einer prozessgesteuerten und künftig zu einer personenzentrierten Versorgung [Blob2002]. Dieser Paradigmenwechsel erfordert eine Neugestaltung der internen und externen Abläufe bei den direkt und indirekt in die Versorgung Einbezogenen.

Als Konsequenz aus dem Paradigmenwechsel erfolgt vielerorts ein Zusammenschluss verschiedener Krankenhäuser zu Klinikketten (stationärer Bereich) und die Einbeziehung der niedergelassenen Ärzte (ambulanter Bereich) sowie eine intensive Kommunikation mit den Kostenträgern (Versicherungen). Dabei stehen die Ärzte und das medizinische Personal als Leistungserbringer sowie die Administration vor der Herausforderung, auch dem Konkurrenz- sowie dem Kostendruck entgegenzuwirken. Das gemeinsame Ziel aller Bemühungen ist das Wohl des Patienten, d.h. die Erhaltung oder Wiederherstellung der Gesundheit. Daher nehmen die Kommunikations- und Informationsprozesse zwischen den kooperierenden Partnern eine zentrale Position in diesen Optimierungsbemühungen ein. Ein gleichartiges Verständnis über die ausgetauschten Daten ist dabei eine unabdingbare Voraussetzung. Gewissermaßen ist das die erste "Definition" von semantischer Interoperabilität.

Im Gesundheitswesen gibt es historisch bedingt eine Vielzahl von Anbietern, die Speziallösungen für administrative und medizinische Dokumentationsaufgaben herstellen und vertreiben. Insbesondere im Bereich der stationären Behandlung hat sich schon früh die Situation ergeben, dass eine bunte Mischung von Systemen in einem Krankenhaus im Einsatz ist. Häufig waren dies Insellösungen, die keinerlei Kommunikation zu anderen Systemen ermöglicht haben. Nachträglich wurden diese Systeme dann um Schnittstellen erweitert, um Daten auszutauschen und damit eine Doppelerfassung und Fehler in den einbezogenen Anwendungssystemen zu vermeiden.

## **1.2. Kommunikation und Kooperation im Gesundheitswesen**

Mit dem zunehmenden Einsatz von Computeranwendungen im Gesundheitswesen steht man vor der grundsätzlichen Anforderung, diese Systeme in die Kommunikations- und Kooperationsketten einzubeziehen und fortgeschrittene Informations- und Kommunikationstechnologien (ICT) für den Datenaustausch zu nutzen. Grundsätzlich gibt es dabei selbst im Rahmen des organisationszentrierten Paradigmas nur zwei Möglichkeiten einer Datenintegration: Entweder installiert man eine Anwendung (und damit Datenbank) innerhalb einer Organisation (zentralisierter Ansatz) oder synchronisiert die Daten über Anwendungen hinweg beispielsweise durch den Austausch von standardisierten Nachrichten<sup>1</sup> (dezentralisierter Ansatz). Der erste Ansatz ist mitunter in einer größeren Organisation wie einem Krankenhaus nur schwerlich zu realisieren, da die einzelnen Systeme vor Stellung der Integrationsfrage angeschafft wurden bzw. für bestimmte Funktionen optimiert sind. Wenn über Organisationsgrenzen hinweg kommuniziert werden muss, dann scheidet dieser Weg grundsätzlich aus, da es keine Anwendung gibt, die von mehreren Dienstleistern gleichermaßen favorisiert wird. Davon unabhängig stehen einem solchen Ansatz datenschutzrechtliche Bedenken entgegen. Als weiteres Argument kommt die Bevorzugung des "Best-of-Breed-Ansatzes" [Hamm2003] hinzu, bei dem man die besten Systeme für den jeweiligen Anwendungsbereich miteinander kombinieren möchte.

## **1.3. Integrierte Versorgung und interoperable Systeme**

Im Gesundheitswesen in Deutschland gibt es eine Reihe von Akteuren, die Daten erfassen bzw. erfasste Daten für ihre Arbeit benötigen (Abbildung 1). Dazu gehören als Leistungserbringer Krankenhäuser (stationäre Versorgung), niedergelassene Ärzte (ambulante Versorgung) und Apotheken sowie als Kostenträger Versicherungen, um nur einige zu nennen. Diese Akteure gehören verschiedenen Bereichen – Sektoren – an. Man spricht deshalb bei einem derart übergreifenden Datenaustausch von intersektoraler Kommunikation.

---

<sup>1</sup> Neben Nachrichten stehen auch andere Paradigmen wie Services zur Verfügung, die aber der hier diskutierten Aufgabenstellung nicht entsprechen und daher nicht weiter verfolgt werden.

Wenn Gesundheitsdienstleister aus verschiedenen Sektoren eng zusammenarbeiten, so nennt man das auch "integrierte Versorgung (IV)".

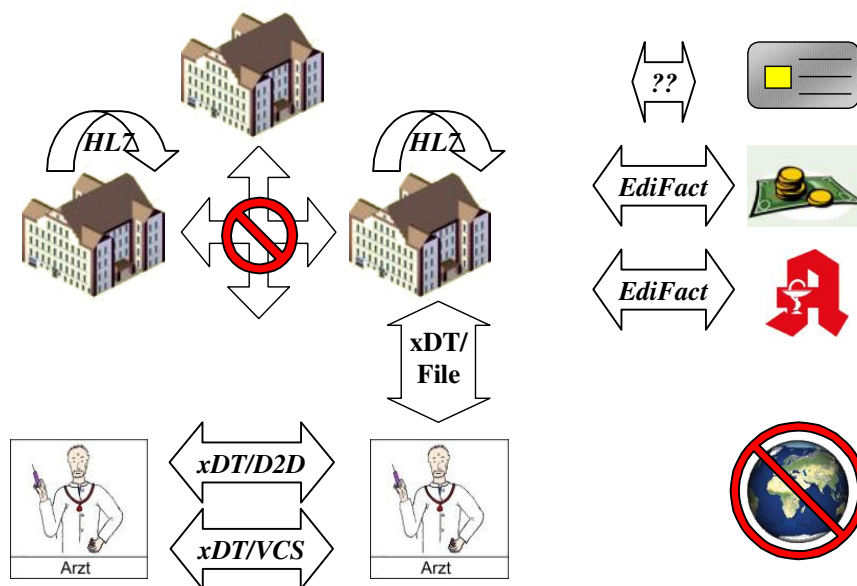


Abbildung 1: Aktuelle Kommunikationsszenarien im Gesundheitswesen

Wie aus der vorstehenden Abbildung hervorgeht, kommen beim Datenaustausch die unterschiedlichsten Kommunikationsstandards zum Einsatz, wobei die im Aufbau befindliche Telematik-Infrastruktur [TI] hier noch nicht berücksichtigt wurde. Welcher Standard konkret verwendet wird, hängt dabei von dem beteiligten Kommunikationspartner ab und hat damit charakterisierenden Einfluss auf den Datenaustausch. So werden vielfach Kommunikationsbeziehungen bilateral abgestimmt und alle damit verbundenen Faktoren direkt miteinander verknüpft. Ermöglicht wird dieses Vorgehen durch die á priori speziell für diesen Datenaustausch festgelegten Intentionen. Durch den sich weiter ausbreitenden Datenaustausch muss von diesem Vorgehen aber Abstand genommen werden. Eine Flexibilisierung ist gefordert, da genau diese Voraussetzungen breit in Frage gestellt werden.

### 1.4. Motivation zur Kooperation

Allerdings sind die Kommunikationsflüsse im informationsgetriebenen Gesundheitswesen weniger trivial; sie sind von einer Vielzahl unterschiedlicher Faktoren geprägt – wie durch die Betrachtung der 4 Seiten einer Nachricht (s. Abb. 2, [SchuvT]) schnell ersichtlich wird. Dies hat implizite Auswirkungen auf die Implementierung von Schnittstellen und die Dokumentation: Der offensichtlichste Aspekt in einer Kommunikation sind die eigentlichen Daten zzgl. der ergänzenden Informationen, die zu übertragen sind, ungeachtet der Form, in der dies geschieht. Normalerweise löst ein Ereignis die Übermittlung einer Nachricht aus, so dass der Appell an den Empfänger in enger Korrelation zu diesem Ereignis steht. Eng verbunden damit ist eine Aussage über das sendende System. Daraus

ergibt sich die Interpretation der Beziehung zwischen diesen beiden Systemen wie beispielsweise die gegenseitigen Erwartungen. [OemPla2006]



Abbildung 2: Die 4 Seiten einer Nachricht nach Schulz von Thun [SchuvT]

Die Durchführung von Kommunikation verfolgt ein Ziel: das Auslösen einer bestimmten Reaktion beim Empfänger. Um dieses Ziel zu erreichen, sind verschiedene Faktoren von Nöten, die allerdings unterschiedlich gewichtet werden können. Je besser beispielsweise die Beziehung zwischen den beiden beteiligten Parteien ist, desto weniger Daten sind notwendig oder desto klarer und unmissverständlicher wird der Appell.

### 1.5. Informationszyklus

Eine andere Betrachtung desselben Szenarios führt prinzipiell zu denselben Ergebnissen: So kann Kooperation zwischen zwei Parteien nur dann erfolgreich verlaufen, wenn sie das gleiche Verständnis über die Prozesse besitzen. Die Übermittlung von Daten (Beobachtung), die unter Anwendung von Wissen interpretiert (verstanden) werden müssen (Information), führt über die Anwendung von Wissen zu entsprechenden Aktionen, die wiederum Ergebnisse produzieren, so dass sich hier der Kreis schließt [Blob2007c].

Gleichgeschaltete Aktivitäten aufgrund bestimmter Daten führen nur dann zu demselben Ergebnis, wenn das gleiche Wissen angewendet wird.

Wenn nun umgekehrt auf beiden Seiten das gleiche Wissen vorhanden ist, dann kommt man mit denselben Daten unter Anwendung dieses Wissens zu demselben Ergebnis. Im Prinzip kann dann von Interoperabilität gesprochen werden.

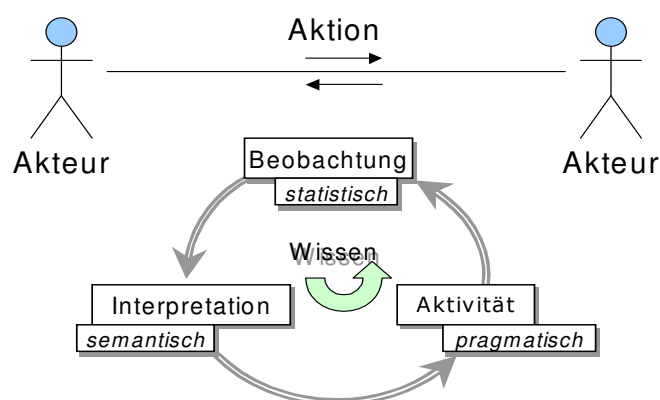


Abbildung 3: Informationszyklus [Blob2007c]

Die verschiedenen, einer bestimmten Phase des Informationszyklus (Abbildung 3) zugeordneten Informationsdefinitionen (statistische, semantische und pragmatische



Interoperabilität) bedingen ein unterschiedliches Ausmaß an Abstimmung zwischen den beteiligten Akteuren [Blob2005, Blob2007b], die, wenn sie eine Vielzahl von Akteuren aus unterschiedlichen Domänen betreffen, in Standards/Normen fixiert werden müssen. Domänen definieren sich durch differenzierende Charakteristika, wie dies durch die Object Management Group (OMG) erfolgte [OMG 1995], oder durch ihre spezifische ontologische Repräsentation [Blob2010]. Die erreichte Harmonisierung in den unterschiedlichen Phasen des Informationszyklus resultiert in unterschiedlichen erforderlichen Interoperabilitätsniveaus der unterstützenden Systeme zur Realisierung der Kooperation. Die Einteilung und Begründung der Standards wird in [Blob2009a] detaillierter betrachtet und in Kapitel 4 noch näher erläutert.

## 2. Aufgabenstellung

Kommunikationsstandards im Gesundheitswesen sind aufgrund verschiedener Paradigmen und Methodologien zueinander inkompatibel. Dies gilt auch für unterschiedliche Standards innerhalb der HL7-Familie („Health Level Seven“, [HL7] ) – namentlich die HL7-Versionen 2.x und 3, welche die Grundlage dieser Arbeit bilden und später detaillierter betrachten werden. Eine Überbrückung im Sinne der Herstellung semantischer Interoperabilität erfordert Wissen, das ausgedrückt und formalisiert werden muss. Bis dato erfolgt dies durch die explizite Programmierung/Konfiguration von Kommunikationsservern.

Die einleitend und nachfolgend vorgebrachten Hintergrundinformationen zum Einsatz verfügbarer Standards zeigen, dass die Erstellung (Entwicklung) eines neuen Programms für den Datenaustausch, das Einhalten der Kodierregeln und die simple Normierung der Datenrepräsentation nicht ausreichen. Innerhalb der obersten Ebene des ISO-OSI Modells [ISO 7498, DIN 7498] - der Anwendungsschicht - gibt es noch weitere Differenzierungen und Fragestellungen, die mit dieser Arbeit untersucht werden sollen.

Die Dissertation stellt deshalb Überlegungen an, ob eine Wissensbasis erstellt und anschließend genutzt werden kann, um die Interoperabilitätsprobleme zwischen HL7 v2.x und V3 zu adressieren und zu lösen. Dies soll anhand einer vereinfachten Repräsentation der Fachdomäne überprüft werden.

Folgende Fragen sollen im praktischen Kontext dieser Arbeit, d.h. ohne den Anspruch einer erschöpfenden philosophischen Fundierung, behandelt werden:

- Wie können die Kommunikationsstandards HL7 v2.x und V3 auf eine bzw. in einer Wissensbasis abgebildet werden?
- Was ist Wissen überhaupt und wie wird Wissen in einer Wissensbasis abgelegt, d.h. wie wird es repräsentiert?
- In welcher Beziehung steht Wissen zu Ontologien?

- Auf welcher Grundlage kann eine derartige Wissensbasis aufgebaut werden?
- Welche Integritätsregeln sind zu berücksichtigen?
- Welche Anforderungen müssen an eine Wissensbasis gestellt werden?
- Kann UML (Unified Modeling Language) als Modellierungstechnik helfen?

### **3. Gliederung der Arbeit**

Die Arbeit ist in mehrere Blöcke unterteilt. Nach der vorangegangenen Motivation der gesamten Arbeit inklusive der Aufgabenstellung werden in einem zweiten größeren Abschnitt weitere tiefer in die Problematik einführende Hintergrundinformationen vorgestellt. Daran schließen sich Definitionen zu für die Arbeit fundamentalen Begriffen an.

Sodann erfolgt die Begründung und Erläuterung der verwendeten, architektur-basierten Methodik, die auf einer Rahmenarchitektur – dem sog. generischen Komponentenmodell (GCM) – beruht. Dabei werden die Grundlagen des GCM erörtert sowie die für diese Arbeit notwendigen Einschränkungen begründet und präzisiert.

Da diese Arbeit in erheblichem Umfang auf der Nutzung von Wissen beruht, muss der Begriff "Wissen" ebenfalls analysiert und formalisiert werden. Diese Formalisierung wird fortgesetzt mit einer Unterteilung/Klassifikation in verschiedene "Repräsentationsformen", die anhand des GCM dargestellt werden.

Der nächste Hauptabschnitt umfasst die Präsentation der Ergebnisse gemäß der Aufgabenstellung. Zunächst werden die beiden Familien von Kommunikationsstandards (HL7 v2.x und V3) analysiert und in ihre Konzepte zerlegt. Auf Basis dieser Modelle werden die entsprechenden Ontologien zur Umsetzung und Realisierung der Wissensbasis hergeleitet und in das GCM eingeordnet.

Die Konsistenz zwischen den HL7-Versionen wird über eine Fachdomäne realisiert, in der die fachlich relevanten Konzepte ontologisch definiert werden. Dieses Modell wird ebenfalls in das GCM eingeordnet und durch einen Mapping-Ansatz mit den HL7-Versionen verknüpft.

Abschließend werden die Ergebnisse vor dem Hintergrund der Aufgabenstellung bewertet. Dies beginnt mit einer Charakterisierung der entwickelten Lösung und der Abgrenzung zu anderen Lösungsansätzen. Außerdem erfolgt eine Bewertung des Nutzens. Ungeachtet der erbrachten Beiträge zur Lösung der Interoperabilitätsproblematik ist mit dieser Arbeit das finale Ziel umfassender Interoperabilität sicher noch nicht erreicht. Daher sollen ein Ausblick auf die nächsten Schritte sowie eine Zusammenfassung die Arbeit abrunden.

Im Anhang findet sich eine Reihe von Zusatzinformationen, die zum Verständnis der Arbeit hilfreich sind und auf die Vorarbeiten hinweisen, die letztendlich zur Erstellung dieser Arbeit insgesamt notwendig waren.

## **B. Grundlagen**

Wann immer zwei Anwendungen bislang Daten miteinander austauschen mussten, haben sich die betroffenen Hersteller auf einen gemeinsamen bilateralen „Standard“ geeinigt, der sowohl die Semantik als auch die Datenrepräsentation festlegt. Typischerweise enthält diese Absprache dann auch nur die Daten, die für den konkreten Anwendungsfall benötigt werden. Auch sind die Kommunikationsszenarien dann genau auf diesen Anwendungsfall ausgerichtet. Auf diese Weise hat sich in der Vergangenheit eine Vielzahl an „Standards“ entwickelt, die jedoch nur bedingt direkt miteinander vergleichbar sind, da sie auf verschiedenen ISO/OSI-Ebenen ([DIN 7498], s.u., vgl. Abb.8) angewendet werden. Daraus ergibt sich, wann sie sich gegenseitig ausschließen bzw. ergänzen.

## **4. Hintergrundinformationen**

### ***4.1. Anforderungen an interoperable Systeme***

Der Datenaustausch zwischen den Anwendungen folgt meist einem einfachen Prinzip: Zu festgelegten Zeitpunkten - das kann sowohl das Auftreten eines Ereignisses in der realen Welt, wie beispielsweise die Aufnahme eines Patienten, als auch ein absoluter Zeitpunkt, wie beispielsweise eine bestimmte Tageszeit sein - wird eine Nachricht übermittelt, mit der alle relevanten Informationen kommuniziert werden (Abbildung 4). Hierbei müssen sowohl Sender als auch Empfänger die gleiche "Sprache" sprechen, um die Dateninhalte zu verstehen:

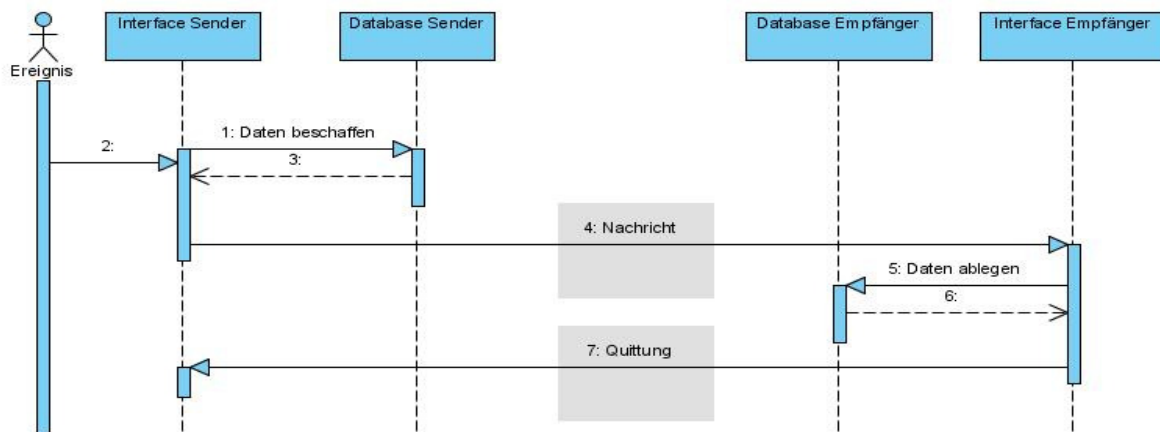


Abbildung 4: Ereignisorientierte Kommunikation

Die Akteure in den verschiedenen Sektoren tauschen hierbei Daten in den unterschiedlichsten Formaten aus (vgl. Abbildung 1). Innerhalb von Krankenhäusern hat sich das im Rahmen des ASTM ([ASTM]) entwickelte und später autonom fortgeführte HL7 als Sprache für den Datenaustausch durchgesetzt und wird inzwischen in allen Ausschreibungen - mitunter auch nur als Schlagwort - gefordert. Für die externe Kommunikation mit den Kostenträgern (beispielsweise die abrechnungsrelevanten Informationen nach §301 GSG SGB V [§301]) werden die Daten jedoch in einer EDIFACT-Variante dargestellt. Die niedergelassenen Ärzte wiederum benutzen untereinander KBV Datenträger-Formate [KBV xDT] wie den Behandlungsdaträger BDT und den Labordatenträger LDT für ihren Datenaustausch, wobei die Absicherung der Datenübertragung inzwischen auf Basis von D2D [D2D, D2D-Impl] oder VCS [VCS] erfolgt.

In einer vereinfachten schematischen Darstellung unter Nennung der existierenden Standards sieht dies wie folgt aus:

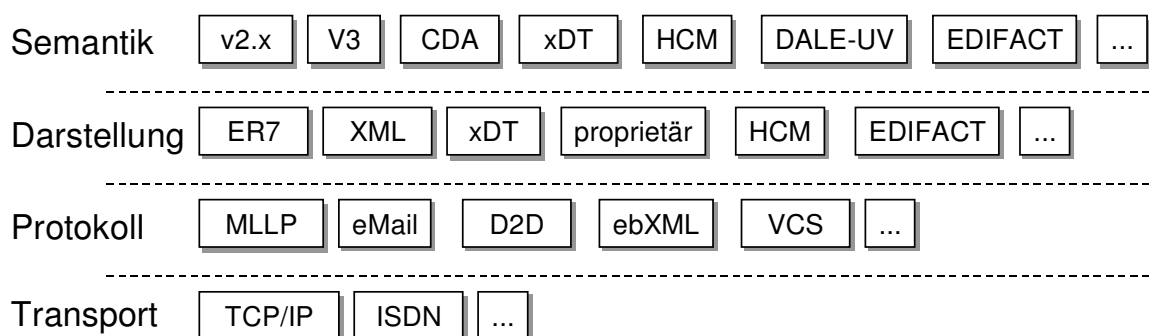


Abbildung 5: Ebenenorientierte Auflistung angewandter Kommunikationsstandards

In obiger Abbildung taucht dieselbe Bezeichnung, bspw. xDT oder HCM, auf unterschiedlichen Ebenen auf. Das heißt, unter ein und demselben Namen wird mitunter sowohl eine Spezifikation für die Semantik/Bedeutung als auch die Darstellung der Daten während

der Übertragung verstanden. Da diese Zusammenhänge nicht allen Beteiligten explizit klar sind, führt dies zu weiteren Problemen bei Schnittstellenverhandlungen.

In dieser ISO-/OSI-Ebenen-ähnlichen Darstellung benutzen die höher liegenden Spezifikationen Dienste der darunter liegenden. Hierbei sind ggf. auch mehrere Zuordnungen möglich, so dass insgesamt eine dreidimensionale Matrix entsteht.

Ein Datenaustausch ist damit genaugenommen zwischen zwei Systemen aber nur dann gewährleistet, wenn beide Seiten auf der obersten, semantischen Ebene denselben Standard benutzen. Die darunter liegenden Ebenen lassen sich relativ leicht über Kommunikationsserver aufeinander anpassen, da es sich hier primär um einfache algorithmische Umsetzungen verschiedener Präsentationsformen und Protokolle handelt. Auf der obersten Ebene sind aber, um das korrekte Verständnis der Inhalte zu gewährleisten, für jede einzelne Information individuelle Aufbereitungen vorzunehmen, die wiederum entsprechendes Wissen erfordern.

## 4.2. Gegenüberstellung der Standards

Im Gesundheitswesen werden neben proprietären Austauschformaten eine Reihe von Standards eingesetzt (s. Kap. 15.2. Ausgelagerte Anhänge). Diese lassen sich unter verschiedenen Aspekten gegenüberstellen.

In der Spalte "SDO" (*Standards Developing Organisation*) bzw. "PEO" (*Profiler Enforcement Organisation*) wird die Organisation genannt, die für die Verabschiedung und Herausgabe zuständig ist. Teilweise handelt es sich hier um private Firmen, so dass die Bezeichnung "Standard" streng genommen nicht gerechtfertigt ist, bzw. es handelt sich um Organisationen, die keine eigenen Standards entwickeln, sondern vorhandene Standards auf eine bestimmte Nutzung (Szenario) anpassen, also Profile erstellen (z.B. IHE).

Aufgrund des Inhalts und der behandelten Domäne ist ersichtlich, dass verschiedene Standards sich einander ergänzen oder direkte Konkurrenten sind.

Tabelle 1: Gegenüberstellung der verschiedenen Standards

Standard <sup>2</sup>	Referenz	SDO/ PEO	Inhalt	Domäne	"Partner"	"Konkurrenz"
HL7 v2.x	[HL7]	ISO + HL7	Patienten- daten	Krh.	DICOM	
HL7 V3	[HL7]	ISO + HL7	Patienten- daten	Gesund- heitswesen		
DICOM	[DICOM]	RSNA	Bilddaten	Krh.	HL7	
IHE	[IHE]	IHE (HL7 +	Pat. + Bilder	Krh.	HL7 + DICOM	

<sup>2</sup> Die Abkürzungen sind im Anhang erläutert.

Standard <sup>2</sup>	Referenz	SDO/ PEO	Inhalt	Domäne	"Partner"	"Konkurrenz"
		RSNA)				
EDIFACT	[UN/EDIFACT a], [UN/EDIFACT b]	CEN	GSG §301	Krh.		HL7
HCM	[SAP HCM]	SAP	Patientendaten	Krh.		HL7
CCOW	[CCOW]	HL7	Synchronisierung Anwendung	Arbeitsplatz	HL7	-
CDA	[CDA]	ISO (HL7)	Dokumente	Krh.	HL7	
SCIPHOX	[SCIPHOX]	HL7 Deutschland	Dokumente	niedergel. Arzt	HL7	
xDT	[KBV xDT]	KBV + ZI	Patientendaten	niedergel. Arzt		HL7
VCS	[VCS]	VdAP	Transport + Patientendaten	niedergel. Arzt	SCIPHOX	D2D (+ HL7)
D2D	[D2D], [D2D-Impl]	KV	Transport + Patientendaten	niedergel. Arzt		VCS
DALE-UV	[DALE-UV]	HVBG	BG-Berichte	D/H-Arzt	VCS + D2D	

### 4.3. Ablauf der Kommunikation

Das Auftreten des Ereignisses nimmt die sendende Anwendung zum Anlass, alle für die Kommunikation benötigten Daten aus der Datenbank herauszusuchen und in einer normierten Form darzustellen. Dazu gehört auch, einzelne Informationen mit Hilfe eines bestimmten Codesystems zu kodieren. Beispielsweise wird das Geschlecht nicht einfach als Zeichenkette "männlich" oder als Ziffer "0" übertragen, sondern in Form eines bestimmten Codes ("M"), dessen Bedeutung, d.h. Semantik, im Voraus mit Hilfe eines Vokabulars abgestimmt wurde.

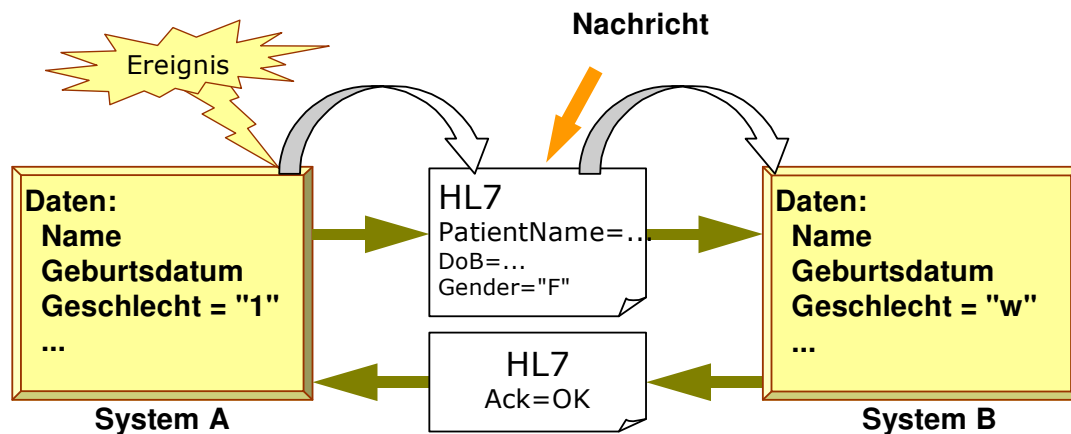


Abbildung 6: Datenaustausch mit HL7

Das empfangende System entnimmt nun die Daten dieser Nachricht, die es für die eigene Arbeit benötigt. Dabei muss es die kodierten Informationen wiederum in das eigene System abbilden.

Die Ebene der Semantik muss genau genommen noch weiter unterteilt werden. In der unteren Hälfte werden allgemeine Konzepte wie beispielsweise Name des Patienten oder Geschlecht definiert. In der oberen Hälfte werden die benötigten Vokabularien festgelegt, also beispielsweise, dass als (vereinfachte) Geschlechtsausprägung "männlich" und "weiblich" zulässig sind.

Neben den beiden bereits erwähnten Ebenen Darstellung und Semantik interessiert noch die Art des Transports, d.h. das Transportprotokoll.

Dieser Hierarchie kann man die ISO/OSI-Ebenen direkt gegenüberstellen [DIN 7498]:

Tabelle 2: ISO/OSI-Ebenen [DIN 7498]

Bereich	Ebene	Bezeichnung	Zuordnung
funktional	7	Anwendung	= Semantik HL7
	6	Präsentation	= Darstellung
	5	Sitzung	= Protokoll
Kommunikation	4	Transport	\
	3	Netzwerk	= Transport
	2	Datenverbindung/Sicherung	/
	1	Physikalisch	/



In der hier durchgeführten Arbeit sind die beiden "primären" HL7-Standards v2.x und V3 die Grundlage des Datenaustausches.

HL7 selbst nimmt für sich in Anspruch, Kommunikation auf oberster ISO-Ebene zu betreiben. Damit ist gemeint, dass man sich über inhaltliche Dinge (Semantik) einigen möchte, unabhängig davon, wie diese dargestellt (Repräsentation) und dann übertragen werden (Transport).

Die Darstellung der Informationen spielt sich auf nächst-niedriger Ebene (Präsentation) ab. Der "HL7-Transport" wird auf Sitzungsebene behandelt, da die Transportebene des ISO-Modells sich mit der Sicherung der Übertragung beschäftigt, welche aber für den "HL7-Transport" schon vorausgesetzt wird.

Während das HL7-Nachrichtenparadigma per Definition applikationsunabhängig ist, fordert die Definition von "Interoperabilität" ([IEEE], vgl. Kap. 5.3. Interoperabilität) als „*the ability of two or more systems to exchange information based on previously agreed upon structure<sup>3</sup> and/or meaning and to use the information that has been exchanged*“ die adäquate Verwendung der Informationen in den beteiligten Anwendungen.

Blobel unterteilt in [Blob2006] und [Blob2007b] Interoperabilität in folgende Stufen (Level), die in Beziehung zum in der Einleitung diskutierten Informationszyklus stehen:

Tabelle 3: Interoperability Levels [Blob2007b]

Interoperability Level	Instances
organizational/service interoperability	common business process
semantic interoperability	advanced messaging, common information model and terminology
syntactic interoperability	messages, clinical documents
structural interoperability	simple EDI, envelopes
technical interoperability	technical plug&play, protocols

Dabei impliziert er, dass die interessierende Interoperabilität die Kooperation der Akteure ist und die o.g. Interoperability-Levels die entsprechenden Anforderungen an die vermittelnden Komponenten beschreiben.

Mead [Mead2005] benutzt eine andere Einteilung, die gerade auf oberster Ebene eine geringere Art der Unterteilung aufweist, aber tendenziell in die gleiche Richtung geht (sinngemäß übersetzt):

**"Syntaktische Interoperabilität"** ist eine Bedingung, unter der Strukturen austauschbar sind. Allerdings gibt es keine Garantie, dass die Semantik über Systemgrenzen hinweg konsistent ist...

<sup>3</sup> Damit wird eine menschliche Interaktion vorab impliziert.

**Semantische Interoperabilität** bezieht sich auf den Austausch von Informationen (Interpretationen von Daten, Bedeutungen) zwischen Systemen.

Die **berechenbare (computable) semantische Interoperabilität** bezieht sich auf den Austausch von konsistenten und eindeutigen Informationen zwischen zwei oder mehr Systemen. Das bedeutet nicht, dass alle Maschinen *a given piece of data* in derselben Art verarbeiten, sondern dass alle Maschinen die Bedeutung von *a given piece of data* **als gleich** interpretieren."

Mit dieser Definition steht Mead im Widerspruch zur IEEE-Definition, die bei der semantischen Interoperabilität schon von einer Nutzungsmöglichkeit der Informationen ausgeht. Er ist weder in der Lage, unterschiedliche Granularitätsniveaus eines Systems zu berücksichtigen, wie es z.B. mittels des GCM in dieser Schrift realisiert wird, noch bildet er den vollständigen Informationszyklus ab. Wirkliche Interoperabilität bedeutet jedoch, wie von Blobel betont, dass mehrere Akteure über den Datenaustausch miteinander kooperieren können, um abschließend ein gemeinsames Ziel zu erreichen.

Beide Definitionen weisen aber auf die Wichtigkeit semantischer Standards wie Vokabularien, Nomenklaturen und Terminologien hin [Blob2006, FreiSchu2009]. Erst diese ermöglichen es, eine Vielzahl von Einzelinformationen in Beziehung zueinander zu stellen. Hierbei ergänzt sich das Expertenwissen beispielsweise mit dem Detailwissen über einen Patienten, so dass sich komplexe Behandlungsabläufe beschreiben lassen, die in ein erweitertes Kommunikationsszenario münden.

Eine zusätzliche Erhöhung der Komplexität wird in diesem Zusammenhang durch die beteiligten Organisationen selbst verursacht, die häufig die Forderung nach einer Anpassung der IT-Systeme an deren Umgebung aufstellen. Diese sind darauf schon durch die Bereitstellung einer Vielzahl von Einzelparametern vorbereitet, die in der Software eingestellt werden können, um das Systemverhalten zu beeinflussen. Diese Parameter haben dann indirekte Auswirkungen auf den Kommunikationsfluss, indem mehr oder weniger Daten bereitgestellt werden, oder sie wirken direkt, indem diese Parameter bei der Kommunikation abgefragt und berücksichtigt werden. [OemPla2006]

Darüber hinaus bieten moderne IT-Systeme ergänzende Konfigurationsmöglichkeiten, so beispielsweise bei minimaler Einarbeitung Formulare für eigene dokumentarische Zwecke zu erstellen und zu nutzen. Das Problem hierbei ist, dass Daten erhoben werden, die im Sinne eines übergreifenden Kommunikationsszenarios jedoch selten weiter verwendet werden können, da die Schnittstellenprogramme nicht wissen, dass diese Daten existieren. Gemäß des bereits einleitend aufgeführten Informationszyklusses (Abbildung 3) fehlt hier zur Ableitung einer Aktivität das Wissen über die Daten, so dass genau genommen nur derjenige, der die Definition des Formulars erstellt hat, dieses auch vollständig nutzen kann – und sei nur zu statistischen Zwecken.

Ziel muss es deshalb sein, dass ein IT-System Wissen über die eigenen Daten (Meta-Daten) erlangt und dieses zur weitergehenden Nutzung verfügbar hält.

## 5. Definitionen

Bevor auf die eigentlichen Details dieser Arbeit eingegangen werden kann, ist eine Reihe von Begriffen relevant, die zunächst erläutert werden sollen:

### 5.1. Modell

Ein Modell wird als eine vereinfachte Darstellung (Abstraktion) von etwas aus der realen Welt verstanden, um das Nachdenken darüber und das Arbeiten damit zu vereinfachen. Hierbei werden zur Vereinfachung alle für die Betrachtung nicht relevanten Details weggelassen. Dieses Vorgehen dient primär der Ableitung neuen Wissens [Blob2006, Blob2009b].

### 5.2. Architektur

Eine Architektur beschreibt ein komplexes System im Zusammenspiel der einzelnen Komponenten, d.h. die Komponenten, deren Funktionen und Relationen [Blob2010].

### 5.3. Interoperabilität

In der Literatur werden mehrere unterschiedliche im Prinzip aber ähnliche Definitionen aufgeführt:

**in-ter-op-er-a-bil-i-ty** (Merriam Webster, [Merriam a]; generische Definition)

"Ability of a system (as a weapons system) to work with or use the parts or equipment of another system."

**interoperability** (IEEE, [IEEE]; IT-bezogene Definition)

"... is the ability of two or more systems or components to exchange information and to use the information that has been exchanged."

**interoperability** (HIMSS, [HIMSS])

"Interoperability means the ability of health information systems to work together within and across organizational boundaries in order to advance the effective delivery of healthcare for individuals and communities."

Die zweite Definition des IEEE ist wohl die bekannteste und am meisten akzeptierte. In der ersten Definition wird die umfassende Kooperation gefordert. In der zweiten Definition stehen zwei Aspekte im Vordergrund: zum Einen die Fähigkeit zum Austausch von Informationen und zum Anderen die Nutzung dieser Informationen.

Der erste Aspekt beschreibt die Herausforderung der syntaktischen Interoperabilität, der zweite die der semantischen mit pragmatischer Konsequenz (Durchführung der intendierten Aktion).

In der dritten Definition (HIMSS) werden in einem Nachsatz sechs Dimensionen der Interoperabilität aufgelistet:

1. *uniform movement of healthcare data (Übertragung)*
2. *uniform presentation of data (Darstellung)*
3. *uniform user controls (Umgang)*
4. *uniform safeguarding data security and integrity (Datensicherheit)*
5. *uniform protection of patient confidentiality (Vertraulichkeit)*
6. *uniform assurance of a common degree of system service quality (Service-Qualität)*

Die daraus abgeleitete Schlussfolgerung ist die Tatsache, dass "Interoperabilität" kein einfach zu beschreibendes Konzept ist. Vielmehr verbirgt sich – angewendet auf den Bereich des Gesundheitswesens – dahinter das Ziel, die Gesundheitsversorgung im Allgemeinen zu verbessern. Die HIMSS-Definition entspricht am ehesten der in dieser Arbeit zugrunde gelegten detaillierten Definition von Blobel [Blob2006, Blob2007b, Blob2010], d.h. der Kooperativität zur Erreichung eines gemeinsamen Geschäftsziels.

Heidenreich reduziert Interoperabilität in [HeiDic2007] auf die „Gleichheit der verstandenen Konzepte“. Der korrekte Umgang mit Konzepten – an dieser Stelle vereinfacht mit Bedeutungseinheiten gleichzusetzen – setzt aber die Einordnung in einen Kontext voraus, so dass bei der Prüfung der Gleichheit der Konzepte auch die Gleichheit der Kontexte sichergestellt sein muss. Diesem letztgenannten Aspekt wird nachfolgend besonders Rechnung getragen.

Heidenreich übersieht hierbei die Tatsache, dass ungleiche Konzepte ebenfalls interoperabel sein können, wenn die Unterschiede bzw. die Ableitungsregeln bekannt sind. Außerdem sichert seine Definition lediglich die Möglichkeit der Interoperabilität, nicht aber deren Praktizierung.

Zusammenfassend lässt sich festhalten, dass umfassende Interoperabilität nur durch einen kompletten Durchlauf durch den Informationszyklus zur vollständigen Abwicklung einer Aktion zwischen den Akteuren sichergestellt werden kann.

## **5.4. Linguistik**

Morris teilt drei zum Wissenszweig der Linguistik gehörenden Begriffe wie folgt ein: [Morris 1938]

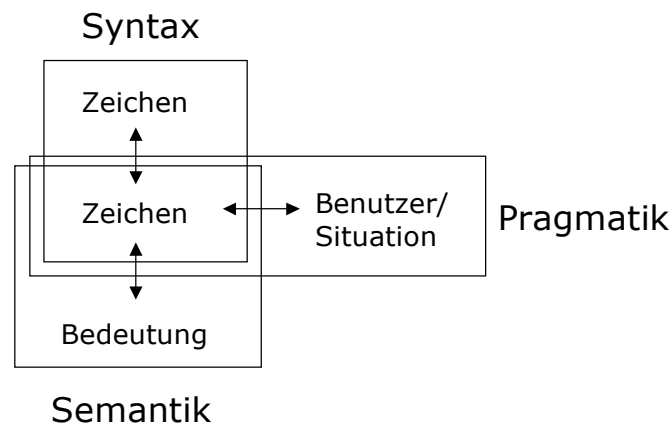


Abbildung 7: Zusammenhang mit Zeichen (nach [Morris1938])

Diese sind für diese Arbeit wichtig sind und sollen nachfolgend kurz erläutert werden.

### 5.4.1. Syntax

"Eine Syntax beschäftigt sich mit der grammatikalischen Struktur von Modellrepräsentationen wie beispielsweise Sprache und Schrift. Sie ist statisch und wird durch eine Grammatik und einschränkende Regeln beschrieben.

Definition: Syntax (griech. *syntaxis* = Zusammenordnung, Lehre vom Satzaufbau). Eine Sprache wird durch eine Folge von Zeichen<sup>4</sup>, die nach bestimmten Regeln aneinander gereiht werden dürfen, definiert. Den hierdurch beschriebenen formalen Aufbau der Sätze oder Wörter, die zur Sprache gehören, bezeichnet man als Syntax." [Duden Inform]

### 5.4.2. Semantik

"The term 'semantics' means meaning or the study of meaning. The meaning of a natural-language word or sentence is the entity or action it denotes." [Hirst]

Hirst geht davon aus, eine "*computational representation for meaning*" zu finden. In seinem Artikel bezieht er sich dabei primär auf natürlichsprachliche Verarbeitung, wie die nachfolgenden nicht-eindeutigen Beispiele zeigen:

"*Nadia washed the dog with long hair.*" [Hirst]  
 „[Ich sehe [den Mann mit dem Fernrohr].]"  
 „[[Ich sehe den Mann] mit dem Fernrohr.]" [Linke]

<sup>4</sup> An dieser Stelle sei noch mal auf die ursprünglich unterschiedliche Darstellung der Zeichen in verschiedenen Zeichensätzen hingewiesen, die mitunter den Datenaustausch zwischen den Applikationen be-/verhindern: ASCII [ASCII], Latin [ISO 8859], Unicode [Unicode], EBCDIC [EBCDIC].

Die Erkennung und Behandlung derartiger Ambiguitäten, von lexikalischen Mehrdeutigkeiten (Polysemie) bis hin zu syntaktischen Ambiguitäten (Anbindung von Nominalphrasen) stellen eine Herausforderung für die korrekte Interpretation sprachlicher Ausdrücke dar.

Aber auch beim Parsen von Sätzen geht es darum, die korrekten Konzepte zu finden und richtig zueinander in Beziehung zu setzen.

### 5.4.3. Semiotisches Dreieck

"Das **semiotische Dreieck** ist ein in der Sprachwissenschaft und Semiotik verwendetes Schema, durch das veranschaulicht werden soll, dass ein Zeichenträger (Symbole) sich nicht direkt und unmittelbar auf einen außersprachlichen Gegenstand bezieht, sondern dieser Bezug nur durch die Vermittlung einer Vorstellung/eines Begriffs erfolgt." [Ogde1923]

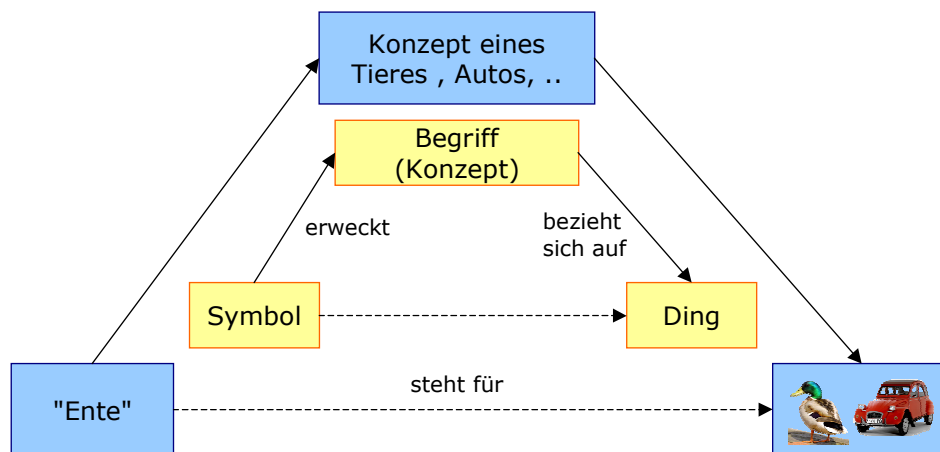


Abbildung 8: semiotisches Dreieck [Ogde1923]

An dieser Stelle sei auf die von Cimino erwähnte Konzeptorientierung und weitere Forderungen an kontrollierte Vokabularien [Cimi1998] verwiesen, über die derartige Zusammenhänge einfacher verwaltet bzw. Probleme vermieden werden können. Unabhängig davon wurde in den 90er Jahren mit der Entwicklung des generischen Komponentenmodells (GCM), das in der Rahmenarchitektur näher erläutert wird, ein analoges und noch umfassenderes Gebäude aufgebaut.

Wie das obige Diagramm andeutet, findet eine Kommunikation über bestimmte Dinge (der realen Welt) statt, welche durch geeignete Konzepte repräsentiert werden. So sind beispielsweise das Auto und das Tier zwei verschiedene Dinge. Ein Konzept muss allerdings auch sprachlich ausgedrückt werden. In diesem Beispiel ist das das Wort „Ente“. In der rein sprachlichen Darstellung ist somit unklar, was genau gemeint ist, wenn dies weder sprachlich präzisiert noch durch den Kontext erschließbar ist.

In dieser Arbeit zur Unterstützung der semantischen Interoperabilität geht es auch darum, von einem Symbol auf exakt ein Konzept zu verweisen/schließen, um Zweideutigkeiten und damit Fehlinterpretationen zu vermeiden.

#### 5.4.4. Pragmatismus

Prof. Dr. Jean-Paul Thommen erläutert im Gabler-Lexikon, dass „Pragmatismus i.w.S. eine weltanschauliche, i.e.S. eine erkenntnistheoretische Position ist, die den Wert von Handlungen oder Erkenntnissen ausschließlich anhand ihres praktischen Nutzens bemisst. Hauptvertreter sind C.S. Peirce, J. Dewey (Instrumentalismus) und W. James. Im erkenntnistheoretischen Bereich läuft der Pragmatismus auf eine Reduzierung von Wahrheit auf (momentane) Nützlichkeit hinaus“. [Thom2010]

#### 5.4.5. Pragmatik

**Prag|ma|tik**, die; -, -en [griech. pragmatike (téchnē) = Kunst, richtig zu handeln, zu: pragmatikós, →pragmatisch] [Duden Uni]

### 5.5. *Ontologie*

Der Begriff der Ontologie stammt genau genommen aus dem Bereich der Philosophie und hat seinen Ursprung in der griechischen Sprache: Ontos (Sein) und Logos (Wort) = Lehre vom Sein [Quine1948].

Eine andere, vorzugsweise in der Informatik verwendete und zum ursprünglichen philosophischen Ansatz im Widerspruch<sup>5</sup> stehende Definition beschreibt eine Ontologie als Bezeichnung für Modelle, die den Anspruch erheben, einen Ausschnitt der Realität inklusive ihrer Strukturen abzubilden.

Oft wird dabei auf die Definition von Tom Gruber verwiesen [Grub1992, Grub1993, Gruber2008]:

*"An ontology is a specification of a conceptualization."*

Gruber bezeichnet Ontologien demnach als "explizite formale Spezifikation einer gemeinsamen Konzeptualisierung (Begriffsbildung)." Das inkludiert gleichzeitig Relationen zwischen den benannten Konzepten – ohne die Art der Relation dabei einzuschränken. Auf diesen Sachverhalt wird später bei Aufbau der eigenen Ontologien noch Bezug genommen. Damit verschwimmt aber auch die Grenze zwischen Ontologien und Wissensrepräsentation.

Die verschiedenen Definitionen werden in einer Vielzahl von Artikeln analysiert und diskutiert (s. Fußnote).

Hinsichtlich der Formalisierung und der Ausdrucksstärke lassen sich die verschiedenen Ontologien in folgende Typen einteilen. (Die ursprüngliche Darstellung stammt von Uschold und Gruninger [UschGru2004]. Nachfolgend die überarbeitete Darstellung.):

---

<sup>5</sup> Zum Begriff der Ontologie und den dazugehörigen Sichten siehe auch [Kusn2006] und [SchuBei2009].

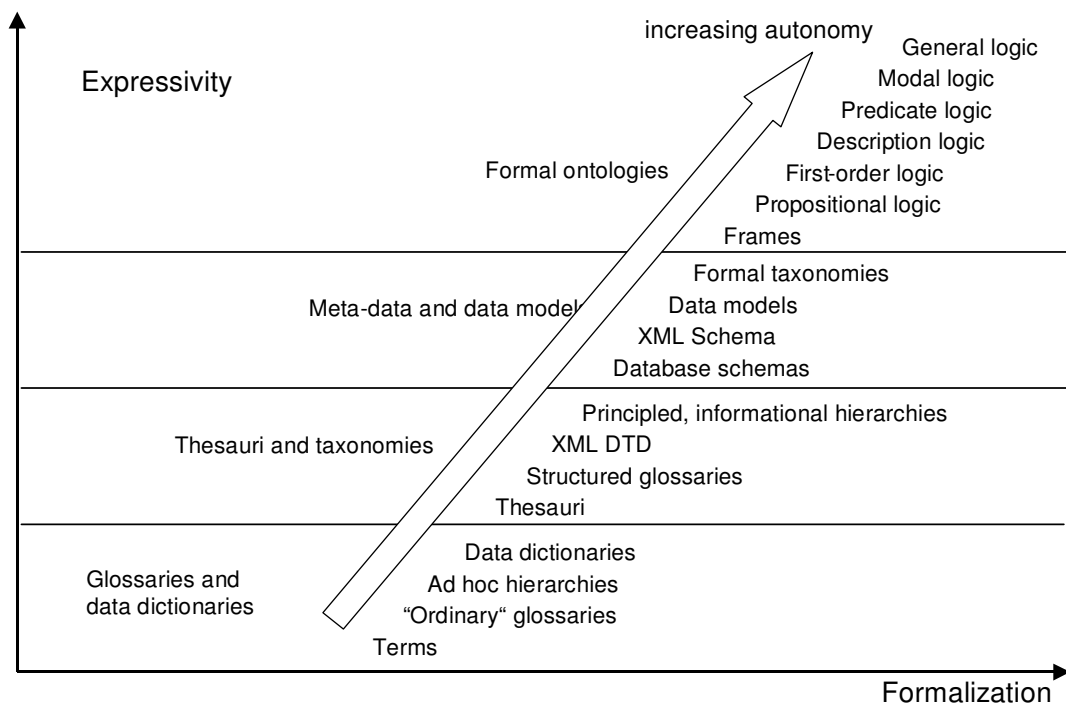


Abbildung 9: Repräsentationsformen für Ontologien [BloOem2009]

Auf der anderen Seite können Ontologien aber auch bezüglich ihrer Komplexität zur Realisierung des Anwendungszwecks unterteilt werden [BloOem2009]:

- generelle (philosophische) Ontologien
- Top-Level-Ontologien
- Domänen-Ontologien
- Anwendungs-Ontologien

Die ICT-Ontologie realisiert die computergestützte Repräsentation der genannten Ontologien [Blob2010].

Diese hierarchische Unterteilung wird im weiteren Verlauf der Arbeit noch näher betrachtet werden.

## 5.6. Epistemologie (Erkenntnistheorie)

Die Epistemologie – oder auch Erkenntnistheorie – bezeichnet in der Philosophie einen Bereich, der sich mit der Natur und der Herkunft von Wissen beschäftigt [Nutt1987, Cocc2007]. Die Basiskonzepte dazu sind Glauben, Wahrnehmung, Darstellung, Rechtfertigung, Beschreibung sowie Auswertung.

Dazu muss der Glauben, der als Wissen genutzt werden kann, qualifiziert werden, um von dem Glauben differenziert werden zu können, der sich dazu nicht eignet. Diese Aufgabe ist mitunter nur schwer zu lösen.

Nach Bodenreider [BodSmi2004] ist "*epistemology ... the study of biological or medical knowledge. Thus it encompasses the ways in which physicians come to know about the existence of given diseases in given patients.*"



## 5.7. Wissen

Wissen drückt im Unterschied zu Glauben – nicht: Gläubigkeit/Vertrauen – einen Sachverhalt aus, der sich begründen lässt und somit auf Fakten (Daten/Informationen) beruht. In der allgemeinen Erkenntnistheorie wird Wissen als "wahre gerechtfertigte Meinung" definiert: In der darüber entstandenen Debatte (z.B. [Gett1963]) wird allerdings an Beispielen demonstriert, dass dies nicht ausreicht, da diese durchaus auch zufällig – also ohne Anwendung von Wissen – erzielt werden kann. David Armstrong fordert deshalb eine weitere Bedingung, nämlich dass die der Meinung zugrunde gelegte Annahme richtig ist [Arms1973].

Aamodt und Nygard [AamNyg1995]) haben mit ihrer Wissenspyramide gezeigt, wie Symbole Daten repräsentieren, die durch Interpretation zur Information werden. Wenn Wissen darauf angewendet wird, lassen sich aus den Daten über Entscheidungen entsprechende Aktionen ableiten. Dabei wurde jedoch vernachlässigt, dass auch die Semantik, d.h. die Interpretation der Daten Wissen erfordert [Blob2007c], wie es im Informationszyklus (Abschnitt 1.5) demonstriert wird.

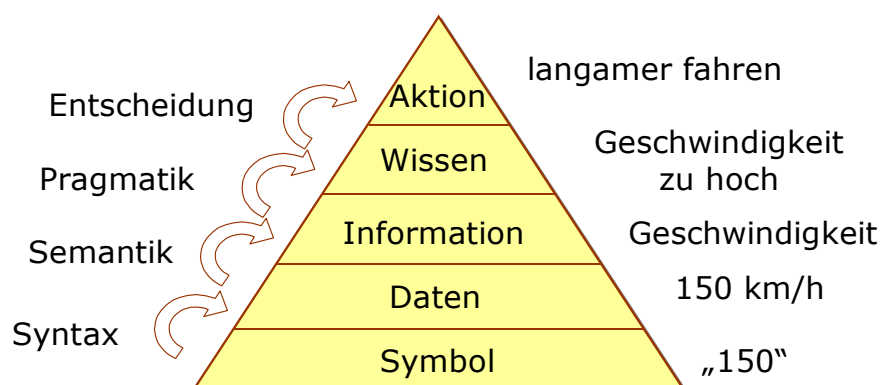


Abbildung 10: Wissenspyramide (nach [AamNyg1995])

Da Aktionen wiederum zu neuen Daten führen, haben wir den in der Einleitung aufgeführten Informationszyklus etabliert.

Wissen kann darüber hinaus noch weiter klassifiziert werden:

explizit vs. implizit: Wissensinhalte, über die man bewusst verfügt und sie ggf. auch formulieren/darstellen kann, gelten als explizit.

deklarativ vs. prozedural: Fakten und Aussagen werden als deklaratives Wissen bezeichnet, während Fähigkeiten als prozedurales Wissen betrachtet werden.

## **5.8. Wissensrepräsentation**

Unter Wissensrepräsentation versteht man den Ansatz, Programme mit Wissen auszustatten [KraMyl1987].

Für die Repräsentation von Wissen über Systeme gibt es eine Reihe verschiedener Möglichkeiten: Entity-Relationship-Modelle, UML-Diagramme, Petri-Netze, semantische Netze und Frames – um nur einige zu nennen. Dieses Wissen wird den Programmen dann entweder prozedural durch explizite Programmierung oder abstrakt in Form eines Interpreteransatzes zur Verfügung gestellt.

## **5.9. Wissensbasis**

Nach [KrayMyl 1987] ist eine Wissensbasis eine Sammlung von Fakten über die reale Welt. Somit ist auch ein Modell eine Wissensbasis [Blob2009b], auch wenn sich bei späteren Verfeinerungen herausstellen sollte, dass dieses Modell Fehler enthält und somit nur spekulativ war.

Umgekehrt ist laut Blobel eine Wissensbasis das Ergebnis eines Prozesses, d.h. die Anwendung einer entsprechenden Methodologie, die wiederum eng an eine geeignete Form der Wissensrepräsentation gekoppelt ist. Dazu müssen wissenschaftliche Aussagen getätigt werden.

## **5.10. Konzept**

Nach Cimino bauen immer mehr Autoren von Vokabularien auf dem Begriff des Konzeptes (s.a. [Smit2004]) auf – "*an embodiment of a particular meaning*". "Bei der sog. Konzeptorientierung muss ein Term genau eine Bedeutung haben und Bedeutungen nicht zu mehr als einem Term gehören." [Cimi1998] (Cimino spricht hier zwar von „Term“, meint gemäß seiner Definition damit aber ein Konzept.)

Damit etabliert sich "ein Konzept" als Grundbaustein von Ontologien, an dem jeweils eine bestimmte Semantik festgemacht wird.

Für Zaiß [Zaiß] ist ein Begriff oder Konzept eine "Denkeinheit, die aus einer Menge von Gegenständen unter Ermittlung der diesen Gegenständen gemeinsamen Eigenschaften mittels Abstraktion gebildet wird". (vgl. auch [DIN 2330]) Diese Definition geht hingegen klar in Richtung einer Klassifikation.

"A concept is a formal model" laut Blobel [Blob2006]. "It shall be uniquely identifiable, accepted by experts and users, as well as independent. A concept as a knowledge component can be specialized and generalized as components can."

Diese doch unterschiedlichen Definitionen verfolgen dieselbe Intention: eine Denkeinheit, die unterschiedliche Granularitäten aufweisen kann. Dabei weist Blobel schon auf die Relationen (s.u.) zwischen verschiedenen Konzepten sowie die Grundlagenbildung für das

generische Komponentenmodell (GCM) hin, so dass diese Definition im weiteren Verlauf der Arbeit zugrunde gelegt wird.

Konzepte müssen nicht unbedingt atomar sein. Die von Zaiß erwähnte Kombination mit Eigenschaften dient ebenfalls diesem Zweck. Wenn konkrete Konzepte detaillierter definiert werden, dann führt dies automatisch zu einer Erhöhung ihrer Gesamtanzahl. Auf diese Weise werden die Konzepte immer granularer, so dass ein bestimmter Sachverhalt dann nur über entsprechende Relationen der Konzepte<sup>6</sup> untereinander ausgedrückt werden kann:

### **5.11. Beziehungen (Relationen)**

Die Nutzung von Konzepten in einem konkreten Szenario ist nur sinnvoll (abgesehen von einer reinen Identifikation), wenn diese jeweils in Beziehung zueinander gesetzt werden. Dabei können Relationen unterschiedlichster Natur auftreten. ISO/FDIS 17115:2007 [ISO 17115] listet hier folgende auf:

- hierarchisch
  - generisch (is-a)
  - partitiv (part-of)
- assoziativ
- sequentiell
  - zeitlich
- kausal

Neben diesen klassischen Relationen sind in der medizinischen Domäne aber auch weitere fachspezifische wie beispielsweise "verursacht durch" oder "Indikation für" denkbar, welche wiederum eine Spezialisierung der bereits erwähnten sein können.

Vom Prinzip her ist eine Relation wie beispielsweise "is-a" ebenfalls ein eigenständiges Konzept – allerdings eines, das mathematisch betrachtet nicht terminal ist, sondern noch Parameter erwartet.

Die Instanziierung einer Relation zwischen zwei individuellen Konzepten ist dann auch ein Konzept, wobei die Parameter konkret festgelegt sind.

Mathematisch betrachtet sind bei Relationen die üblichen relationalen Gesetzmäßigkeiten wie Injektivität, Surjektivität, Bijektivität aber auch Inversion und Transitivität zu berücksichtigen und für die jeweiligen Relationen gesondert festzulegen, z.B. Transitivität der allgemeinen part-of-Relation.

---

<sup>6</sup> Dies ist auch der Ansatz für die Prä- und Postkoordination, bei der dies implizit oder explizit vorgenommen wird.

## 5.12. Klassifikation

"Ziel der Klassifikation ist die Ordnung der Grundgesamtheit durch das Erkennen von Gemeinsamkeiten von Objekten und die Erkennung von Mustern." [Blob1987, JakÜst2007]

Klassifikationen sind Ordnungssysteme, die auf dem Prinzip der Klassenbildung beruhen. In einer Klasse werden alle Dokumentationsobjekte zusammengefasst, die in mindestens einem (klassenbildenden) Merkmal übereinstimmen. Eine Klassifikation ist in diesem Zusammenhang das Ergebnis einer Einteilung (Ordnungssystem) von Objekten anhand von Merkmalen in Gruppen, um eine Vergleichbarkeit zu erleichtern. Das klassenbildende Merkmal sollte sich an der Fragestellung orientieren. [Zaiß]

Die Klassen sollen das zu dokumentierende Gebiet vollständig erfassen. Die Klassenbildung hat mathematisch-logischen bzw. statistischen Regeln zu folgen, wie der Forderung nach Disjunktheit, Ein- bzw. Ausschlusskriterien, Gleichbesetzung der Klassen, Orthogonalität, etc. Dazu dienen Klassifizierungsregeln.

Für Klassifikationen werden meistens hierarchische Ordnungen genutzt, wobei auch mehrere Achsen (Dimensionen, Kategorien) möglich sind (z.B. beim TNM-System) [Gräb2009]:

Damit einhergehend stellt sich die Frage, was eine "Grundgesamtheit" ist? In der Mathematik sind das Variablen, in der Informatik Begriffe. Ein Begriff ist wiederum eine sehr komplexe Struktur, mit der Eigenschaften eines Objektes der realen Welt zusammengefasst werden. Eine Begriffsbildung impliziert dazu aber Wissen. Folglich ist die Klassifikation von Begriffsmengen der Versuch der Ordnung von Wissen. [Blob1987]

Bei nicht vollständiger Präzisierung ergibt sich hier aber das Problem, dass einigen Klassen keine eindeutige Semantik zugeordnet werden kann (Catch-all). Bei dieser vergrößernden Klassifikation gehen wichtige Details verloren [Haas2005, Gräb2009]. Beispielsweise ist dies bei der internationalen Klassifikation für Krankheiten (ICD, [DIMDI ICD, WHO]) der Fall, weshalb über das Deutsche Institut für Medizinische Dokumentation und Information (DIMDI) die sog. AlphaID [DIMDI AlphaID] entwickelt und zur Verfügung gestellt wird. Von ID Berlin wird aus dem gleichen Grund die proprietäre ID-MACS [ID Berlin] favorisiert.

Das nachfolgende Beispiel für diverse Alpha-IDs zur Differenzierung eines einzelnen ICD-Codes verdeutlicht diesen Informationsverlust:

Tabelle 4: Klassifikation: Alpha-IDs und ICD-Codes

Alpha-ID	ICD-Code	Beschreibung
----------	----------	--------------

Alpha-ID	ICD-Code	Beschreibung
I23245	A41.9	Allgemeininfektion
I22163	A41.9	Bakterielle Toxämie
I110208	A41.9	Biliäre Sepsis
I22164	A41.9	Blutvergiftung
I21827	A41.9	Eitervergiftung
I80639	A41.9	Gangränöse Sepsis
I69788	A41.9	Generalisierte Eiterabsorption
I73855	A41.9	Generalisierte Infektion a.n.k.
I80668	A41.9	Kryptogenetische Sepsis
I111363	A41.9	Postoperative Sepsis
I21828	A41.9	Pyoseptikämie
I21825	A41.9	Pyämie
I69798	A41.9	Pyämische Embolie
I111216	A41.9	Sepsis bei unbekanntem Primärherd
I21826	A41.9	Septikopyämie
I22168	A41.9	Septikämie
I111145	A41.9	Septisch-toxischer Schock
I32814	A41.9	Suppurative Sepsis
I22166	A41.9	Toxikämie
I22165	A41.9	Toxämie
I32854	A41.9	Urosepsis

Folglich muss eine Klassifikation möglichst feingranular sein, damit sie für eine klinische Dokumentation eingesetzt werden kann [Haas2005].

## C. Methoden

In diesem Hauptkapitel werden zuerst die Grundlagen dargestellt, auf denen die Inhalte dieser Arbeit anschließend aufgebaut werden. Nach der bereits erfolgten kurzen Definition, was ein "Modell" überhaupt ist, wird das Referenzmodell für verteilte Verarbeitung (RM-ODP) auf Brauchbarkeit überprüft und daraus Schlussfolgerungen für die zu verwendende Referenzarchitektur abgeleitet.

Die nachfolgende Auflistung orientiert sich hierbei an der Komplexität der zugrunde liegenden Struktur und der jeweils damit realisierbaren Modelle. Historisch sind diese jedoch in der Reihenfolge GCM, RM-ODP und SAEAF/SAIF entstanden, wobei das GCM auch den beiden anderen konzeptionell zugrunde liegt.

## 6. Rahmenarchitektur

Das dieser Arbeit zugrunde liegende Thema enthält eine Reihe von Schwierigkeiten, die durch Bezugnahme auf eine Rahmenarchitektur vereinfacht und in Beziehung zueinander gesetzt werden können. Wie nachfolgend dargestellt wird, hat sich gezeigt, dass das generische Komponentenmodell (GCM, [Blob1997, Blob2000, Blob2002, Blob2007b]) die notwendige Funktionalität zur Verfügung stellt und somit den für diese Arbeit passenden Rahmen darstellt.

Das später näher erläuterte GCM besitzt drei Dimensionen, von denen eine sich mit dem Entwicklungsprozess auseinandersetzt. Dieser wird durch das RM-ODP strukturiert dargestellt:

### ***6.1. RM-ODP –Reference Model for Open Distributed Processing***

Ein Referenzmodell wird als Modell verstanden, das als Ausgangspunkt für die Entwicklung auf konkrete Aufgabenstellungen bezogener Problemlösungen dienen kann. Genau in diesem Licht sieht sich auch das Referenzmodell ODP: es soll Grundlage und Rahmenwerk sein [RM-ODP].

Das RM-ODP definiert dazu grundlegende Konzepte, die für die IT-Systementwicklung in einer Komponentenumgebung benötigt werden. Somit bietet das Referenzmodell ODP auch einen Rahmen für zukünftige Standardisierungen.

Im ersten Teil der Spezifikation (ISO/IEC 10746-1: Overview, [RM-ODP]) findet sich ein Überblick über die 5 verschiedenen Sichten, die untereinander konsistent sein müssen und auseinander abgeleitet werden können:

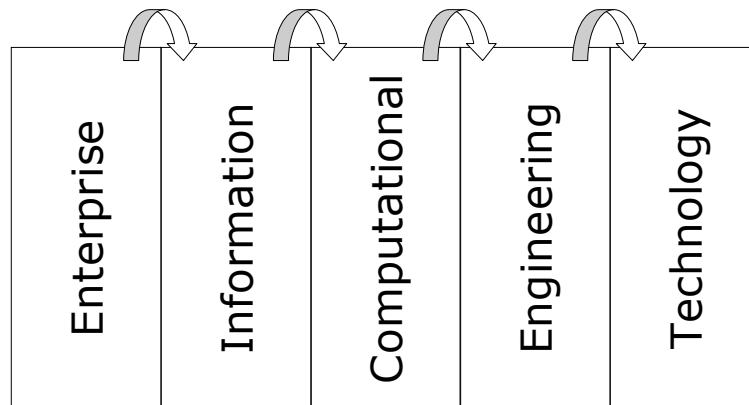


Abbildung 11: Die Sichten des RM-ODP

Die verschiedenen Sichten im RM-ODP sind (nach ISO 10746-1) wie folgt spezifiziert:

Eine Sicht (auf ein System) ist eine Abstraktion, die zu einer Spezifikation des Gesamtsystems in Bezug auf bestimmte Aspekte führt. Fünf Sichten wurden ausgewählt, die sowohl einfach als auch vollständig sind, und alle Phasen des Systemdesigns abdecken. Diese fünf Sichten sind:

- Der *Enterprise Viewpoint* für den Zweck der Definition des Anwendungsbereichs und der Policies, die die Aktivitäten des spezifizierten Systems bestimmen,
- der *Information Viewpoint* mit den Informationen und den Einschränkungen in der Verwendung sowie die Interpretation der Informationen,
- der *Computational Viewpoint* zur funktionalen Zerlegung des Systems in eine Menge von Objekten, die über Schnittstellen interagieren – zur Ermöglichung der Verteilung des Systems,
- der *Engineering Viewpoint* für die notwendige Infrastruktur zur Unterstützung der Verteilung des Systems, und
- der *Technology Viewpoint* zur Auswahl der Technologie für die praktische Implementierung.

Für jede Sicht gibt es eine Sprache, die die Spezifikation des Systems für diese Sicht erlaubt. Die Konzepte zur Objektmodellierung bilden eine gemeinsame Basis für diese Sprachen und ermöglichen die Identifikation von Relationen zwischen den verschiedenen Sichten sowie die Zusicherung der Übereinstimmung zwischen den verschiedenen Darstellungen des Systems in den verschiedenen Sichten.

In der Enterprise-Sicht werden der allgemeine Sinn und Zweck des Gesamtsystems beschrieben. Dazu gehört insbesondere auch die Beschreibung der grundlegenden Prozesse und Regeln. In der Information-Sicht werden die Informationen und die Semantik der konkret zu verarbeitenden Informationen in ihrem funktionalen Verhalten (invariantes, statisches und dynamisches Schema) festgelegt. Die drei sich daran anschließenden Sichten – Computational, Engineering und Technology - dienen der Beschreibung des konkreten Systemverhaltens einschließlich der für eine Implementierung benötigten technologischen Festlegungen.

Der Gegenstand dieser Dissertation ist die Analyse der zugrunde liegenden Kommunikationsstandards und die Verbesserung der semantischen Interoperabilität. Da konkrete Entwicklungen interoperabler Systeme die Festlegung des Systemverhaltens und der interessierenden Informationsobjekte erfordern, ist die Arbeit im Bereich der Enterprise und Information Views einzuordnen. Allerdings reicht die mit dem RM-ODP gebotene Systematisierung zur weiteren Differenzierung im Bereich dieser Views nicht aus. Die gestellten Anforderungen werden mit dem generischen Komponentenmodell erfüllt, das über das RM-ODP hinausgeht.

## **6.2. SAEAF – Service Aware Enterprise Architecture Framework**

Mit SAEAF<sup>7</sup> [SAEAF] versucht die HL7-Community, sich ebenfalls dem RM-ODP zu nähern. Der Hintergrund dabei ist eine weitergehende Formalisierung der Begriffe "Conformance" und "Compliance"<sup>8</sup>, um praktische Interoperabilität (workable interoperability) sicherstellen zu können. Hierzu wurden folgende Schlüsselkomponenten identifiziert:

- die speziellen Artefakte, die notwendig sind, um den Entwicklungsprozess zu spezifizieren,
- das formale Konformanz/Compliance Rahmenwerk für die quantitative Messung der Compliance einer bestimmten Implementierung in Bezug auf eine vorgegebene Spezifikation,
- ein formales Einschränkungsmuster, dass die Überprüfbarkeit von den Anforderungen über die Analyse und das Design bis hin zu den Auswirkungen garantiert, und

---

<sup>7</sup> Dieses Projekt wurde 2010 umbenannt in „*Service Architecture Interoperability Framework*“: SAIF.

<sup>8</sup> Unter "Conformance" wird die Übereinstimmung einer Implementierung mit einer Vorgabe verstanden. "Compliance" ist die Übereinstimmung von Vorgaben hinsichtlich unterschiedlich starker Einschränkungen, d.h., eine Vorgabe ist eine Einschränkung der anderen. Unter "Compatibility" wird zuletzt die gegenseitige Verträglichkeit von Implementierungen oder Vorgaben verstanden. (Weitere Details dazu finden sich in Proposal #605 für HL7 v2.8 [HL7-v2 Prop.DB, HL7-v2 Prop 605].)



- das kontrollierende Rahmenwerk, das etabliert werden muss, um eine effektive Compliance mit- und als Ergebnis aus- den Enterprise Architecture Services für HL7 sicherzustellen.

Damit sollen sich dann die notwendigen Schritte ableiten lassen, um gemäß nachfolgender Grafik (Abbildung 12) einen Datenaustausch zwischen zwei Akteuren zu veranlassen.

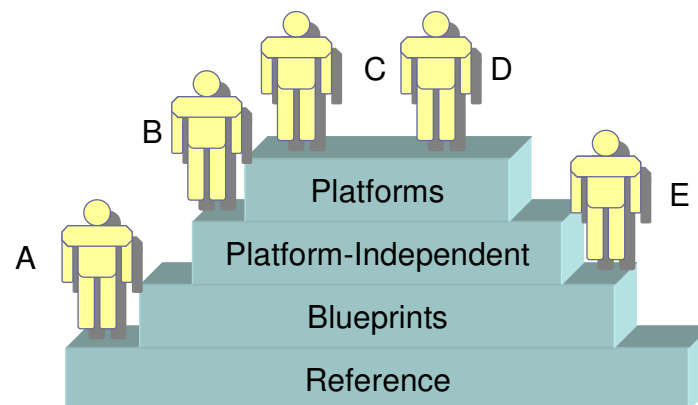


Abbildung 12: "Incremental Approach to Working Interoperability"

Grundlage der Analysen ist dabei das RM-ODP, allerdings müssen hier weitere Unterteilungen vorgenommen werden. Bei SAEAF wird dies als zusätzliche, leider nicht vollständig orthogonale Unterteilung in Analyse, Logik, Implementierung und Technologie realisiert.

Ziel ist dabei zwischen Conformance zu einer Architektur und einem Standard zu unterscheiden.

Mit diesen zusätzlichen Unterteilungen wird deutlich, dass der RM-ODP-Ansatz nicht mehr ausreicht und darüber hinausgehende Modelle eingesetzt werden müssen.

Ein Mangel des SAEAF-Ansatzes ist die Befangenheit im ursprünglichen Nachrichtenparadigma einschließlich der Referenz zum RIM und das Vermissen einer sauberen Architekturdefinition. Daraus resultiert eine parametrische Beschreibung des Verhaltens der kommunizierenden und kooperierenden Systeme in Attributen anstelle der strukturellen und funktionellen Modellierung der in Frage kommenden Anwendungen (s.a. [BloOem2009, Blob2009c]).

### **6.3. Das generische Komponentenmodell**

Die Modellierung von Systemen ist im Allgemeinen zu komplex, als dass alle relevanten Aspekte auf einmal betrachtet werden können. Deshalb versucht das nachfolgende Modell nach dem Prinzip "Vereinfachung und Fokussierung" die Komplexität auf das notwendige Minimum zu reduzieren. Hierbei kann man die Reduktion der Komplexität je nach Betrachtungswinkel in drei verschiedenen Dimensionen durchführen, die unterschiedliche Constraints (Einschränkungen) realisieren.

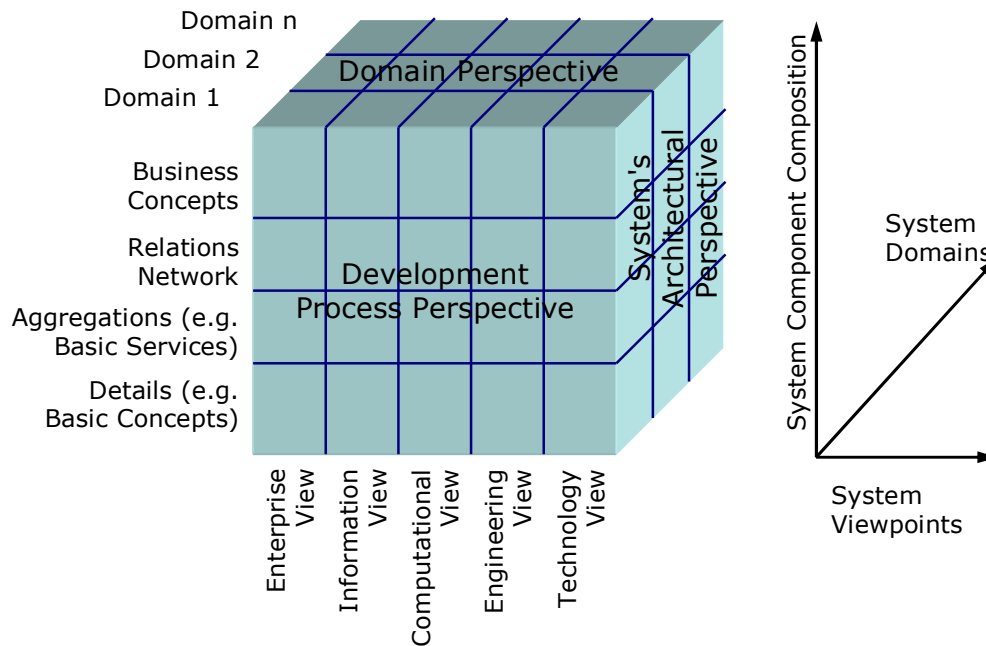


Abbildung 13: Das generische Komponentenmodell [Blob2002, Blob2007b]

Als erstes wären die verschiedenen Domänen zu nennen. Die Klassifizierung von Domänen kann entweder über die Charakteristika der Domänen oder über die Repräsentation ihrer Konzepte realisiert werden [Blob2010]. Dem Ansatz dieser Arbeit folgend, wird der zweite Weg der Differenzierung der Domänen nach den jeweiligen Domänenontologien nachgegangen. Die besagte Einschränkung kann auf die Zuordnung beispielsweise in medizinische, administrative, technische und rechtliche Domänen vorgenommen werden, so dass für eine Betrachtung nur die jeweils zugehörigen Konzepte zu betrachten sind.

Innerhalb einer Domäne können die dann vorhandenen Konzepte, d.h. Entitäten und einbezogene Prozesse, im Sinne einer Architektur in ihre Bestandteile zerlegt oder höherwertige Konzepte aus einfachen Bausteinen zusammengesetzt werden. Damit können die Konzepte auf verschiedenen Granularitätsstufen (granularity levels) betrachtet und im Detail beschrieben werden. Man spricht hier von Komposition und Dekomposition. Die bekanntesten Relationen sind hier die Spezialisierung/Generalisierung im Sinne von Diskrimination/Dekomposition bzw. Aggregation/Komposition sowie die Assoziation (Komponentenrelation).

Folgende Granularitätsstufen werden definiert:

- Geschäftsvorfälle (business concepts)
- Beziehungsnetzwerk (relations network) der beteiligten Entitäten und bezogenen Prozesse
- Aggregationen (aggregations: e.g. basic services/functions)
- Details (details: e.g. basic concepts, transactions)

---

Die dritte Dimension orientiert sich am ISO Referenzmodell "Open distributed Processing" [ISO 10746]. Der Enterprise View beschäftigt sich mit den Geschäftsvorfällen<sup>9</sup>, der Information View konzentriert sich auf die zu verarbeitenden Informationen, während der Computational View sich um die Algorithmen und Dienste kümmert. Diese drei Sichten werden durch plattformunabhängige Modelle beschrieben, die sich allein auf die Logik konzentrieren.

Wie eine konkrete Umsetzung auf eine bestimmte Plattform vorgenommen wird, beschreibt die Engineering View. Die Technology View repräsentiert technische und organisatorische Implementierungsaspekte.

Das zu beschreibende System kann nun durch Aggregation der unter den o.g. Aspekten beschriebenen Einzelkomponenten - den Konzepten - zusammengesetzt werden. Die Beziehung der Konzepte untereinander wird durch Regeln überwacht, die ihrerseits wieder eigene Konzepte darstellen.

Die Funktionsfähigkeit dieses Ansatzes kann nur dann garantiert werden, wenn alle Konzepte - also die Komponenten und deren Wechselbeziehungen - eindeutig identifizierbar sind.

Zusammengefasst beschreibt die Referenzarchitektur des GCM die Komponenten eines Systems, ihre Funktionalitäten und Wechselbeziehungen durch die zugeordneten Konzepte als entsprechend der Zielstellung und Intention des Architekten vereinfachte Modelle der Realität, wobei das System unter den Gesichtspunkten verschiedener Domänen in Subsysteme zerlegt und diese separat betrachtet werden können. Dabei ist festzuhalten, dass die Komponenten in ihrer Struktur und Funktion zu beschreiben sind und Constraints unterworfen werden können, die strukturelle und funktionelle Flexibilität vor allem aber aus den Interrelationen der Komponenten resultieren. Da die Wechselbeziehungen zwischen den Komponenten und ihren Konzepten nicht nur innerhalb sondern auch zwischen den Domänen wirken, müssen für das Design des realen Systems alle Domänen zusammengeführt werden. Die Summe der Konzepte sowohl der Komponenten als auch der Beziehungen zwischen ihnen innerhalb einer Domäne beschreibt die Domänenontologie, wobei es pro Domäne eine oder mehrere Ontologien gibt. Letzteres sollte dazu führen, die Domäne in die entsprechenden Subdomänen so zu unterteilen, dass schließlich eine Domäne durch eine Domänenontologie repräsentiert wird. Die Verknüpfung der Domänen erfordert eine Verknüpfung der Ontologien über Referenz- oder Meta-Ontologien. Eine Systemarchitektur muss über die formale Beschreibung der Konzepte und Regeln definiert werden und somit ontologiegetrieben sein.

---

<sup>9</sup> Aus der Sicht der Interoperabilität sind alle Vorfälle Geschäftsvorfälle (Aktionen im gemeinsamen Interesse).

Die generische Natur des GCM erlaubt seine Anwendung auf alle Systeme von technischen über biologische bis hin zu Sprachen oder abstrakten Strukturen wie Ontologien. Dabei kann das GCM zur Entwicklung von Ontologien benutzt werden, oder Ontologien werden benutzt, um Systeme aus bestimmten Domänen zu beschreiben. Die Dissertation beschränkt sich auf den letzteren Fall.

### **6.3.1. Basisprinzipien**

Das GCM folgt drei Grundprinzipien. Es ist:

- komponentenorientiert
- modellgetrieben
- serviceorientiert

Für diese Arbeit ist die Komponentenorientierung am wichtigsten, da nur auf diese Art die unterschiedlichsten Aspekte und Sichten in Teile zerlegt und unabhängig voneinander beschrieben werden können. Deshalb wird im weiteren Verlauf zuerst eine Dekomposition in verschiedene Domänen vorgenommen, die abschließend wieder zusammengeführt werden. Damit letzteres überhaupt möglich ist, muss es zur Orientierung eine übergeordnete Sammlung von Konzepten geben. Im Sinne der Komponentenorientierung ist das die übergeordnete, weniger granulare Komponente, die quasi als Referenzarchitektur fungiert, von der die darunter liegenden Spezialisierungen abgeleitet werden.

Das zugrunde legen von Modellen, die in einer gemeinsamen Sprache ausgedrückt werden, ermöglicht die Konsistenzprüfung und damit -sicherung.

Grundsätzlich ist es irrelevant, in welcher Form die Konzepte ausgedrückt werden. Formale Sprachen erleichtern jedoch die Verifikation, ob die Paradigmen eingehalten werden, und damit die Sicherung des Verstehens.

### **6.3.2. Vorteile**

Die bisher beschriebenen Eigenschaften des GCM führen zu folgenden Vorteilen [BloEng2006]:

- Offenheit
- Flexibilität
- Skalierbarkeit
- Portabilität
- Benutzerfreundlichkeit
- Verteilung der Komponenten
- Service-Orientierung
- Standardbasiertheit
- Interoperabilität auf den in Tabelle 3 definierten verschiedenen Niveaus

## 6.4. Einschränkung für diese Arbeit

Der Hauptgegenstand dieser Arbeit ist die Betrachtung unterschiedlicher Familien von Kommunikationsstandards. Hier werden HL7 v2.x und V3 als Stellvertreter für zwei zueinander inkompatible Standards verwendet, um die Lösung der damit verbundenen Interoperabilitätsprobleme auf Basis von OWL zu demonstrieren. Jede Version eines Kommunikationsstandards entspricht hierbei einer eigenen Domäne, so dass sich auch Inkompatibilitäten innerhalb einer Familie bewältigen lassen.

CDA Rel.2 [CDA] als Dokumentenformat basiert vollständig auf HL7 V3, da das entsprechende Domänenmodell (D-MIM) nach der HL7-Methodologie aus den RIM-Klassen aufgebaut ist. Die Einbeziehung von CDA Rel.2 in diese Arbeit wäre somit problemlos möglich. (Im Gegensatz dazu ist CDA Rel.1 eine noch nicht harmonisierte Fassung, so dass sie als eigenständiger "Kommunikationsstandard" behandelt werden müsste.)

Die Umsetzung der zugrunde gelegten Informationen über die verschiedenen Views (Enterprise, Information, Computational, Engineering und Technology) ist nicht Bestandteil dieser Arbeit und wird deshalb nicht weiter betrachtet. Siehe hierzu z.B. [Lopez2008].

Das GCM lässt sich zur Vereinfachung der Modellentwicklung einschränken. Die resultierenden Schichten – das sind der Enterprise View (für die Geschäftsvorfälle) und der Information View (für die Daten) – sind in Abbildung 14 farblich hervorgehoben:

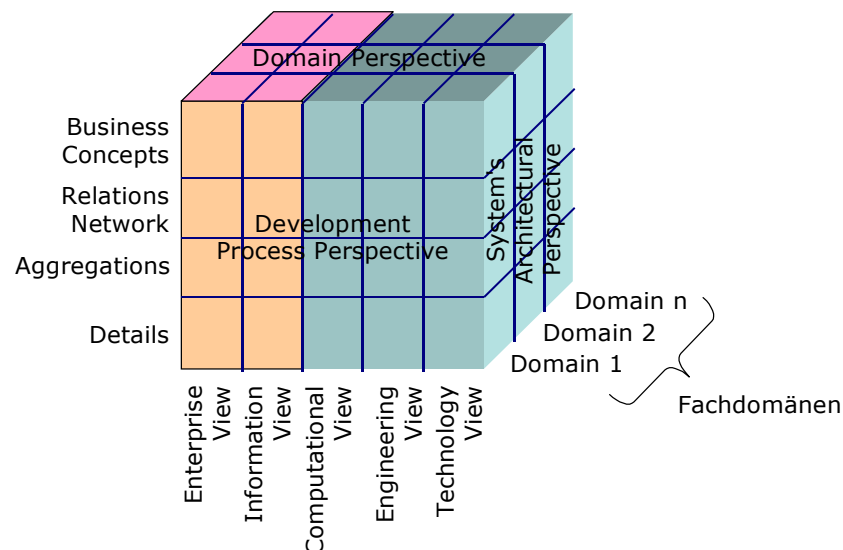


Abbildung 14: Einschränkung des generischen Komponentenmodell [Blob2002, Blob2007b]

Es bleibt die Frage nach den Fachdomänen und deren Beziehungen zueinander.

Die Kommunikation zwischen zwei Systemen (Anwendungen) erfolgt traditionell auf Basis des Austausches von Nachrichten. Es sind aber auch andere Paradigmen wie Dokumente und Services denkbar. Wie bereits in den Einschränkungen zu dieser Arbeit ausgeführt, lässt sich CDA Rel.2 direkt mit abdecken. Grundsätzlich geht es primär um die

Modellierung des Inhaltes der Nachrichten, so dass die Übertragung auf die anderen Paradigmen problemlos möglich ist.

Die internen Architekturen von Anwendungen unterscheiden sich grundlegend von Hersteller zu Hersteller. Prinzipiell sind die einzelnen Anwendungen zur Unterstützung eines (klinischen) Arbeitsablaufes (beispielsweise Untersuchungen von Proben im Labor oder die Patientenverwaltung zwecks Abrechnung) entwickelt worden. Sie können durch die 4 Ebenen des GCM abgedeckt werden, wobei jeweils eine geeignete Systemdefinition vorgenommen und die resultierende intendierte Granularität bzw. Komplexität der Betrachtung fixiert werden muss. In der nachfolgenden Grafik (Abbildung 15) ist dies durch die beiden grauen Kästchen dargestellt. Diese Anwendungen sollen nun ihre Informationen miteinander austauschen. Auf Basis von HL7 – die Unterscheidung zwischen v2.x und V3 ist dabei irrelevant – werden hierbei keine Geschäftsvorfälle beschrieben, so dass die obere Ebene nicht abgedeckt werden muss. Die drei unteren Ebenen sind in den Nachrichten (symbolisiert durch die schwarzen Pfeile) nur implizit abgedeckt. Dabei werden die Konzepte, d.h. das Verständnis von Informationen innerhalb einer Anwendung, nicht direkt kommuniziert, sondern über Vorverhandlungen zwischen den Beteiligten abgestimmt. Dies wird in der nachfolgenden Darstellung durch den Mediator in der Mitte ausgedrückt.

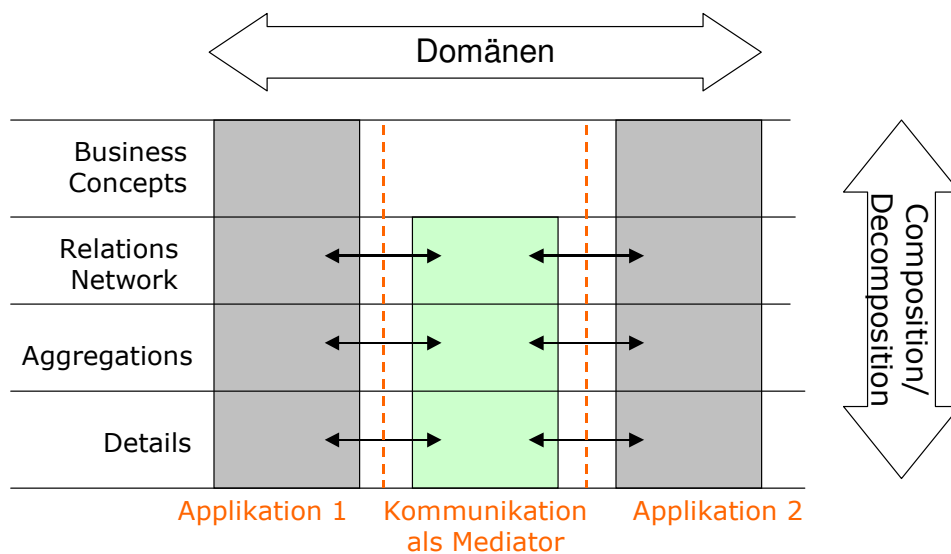


Abbildung 15: Grundlagen der Fachdomänen

Dieser Mediator wird beim Nachrichtenaustausch im realen Einsatz typischerweise als Kommunikationsserver bezeichnet. Über eine manuelle Konfiguration wird dieser in die Lage versetzt, zwischen zwei Anwendungen zu vermitteln, indem die Daten "geeignet übersetzt" werden. Hierzu müssen die Hersteller beitragen, indem sie Informationen über ihr Verständnis der Daten und der Darstellung in einer Nachricht bereitstellen. Eine darauf aufbauende Analyse der syntaktischen und semantischen Unterschiede erlaubt dann eine explizite Anpassung.

Rein pragmatisch wird in der Praxis allerdings anders herum vorgegangen. Da die Schnittstellen auf Kommunikationsstandards basieren, wird davon ausgegangen, dass die jeweiligen Interpretationen korrekt sind. Das heißt, eine Anpassung der Konfiguration wird nur in dem Fall vorgenommen, in dem Daten verfälscht oder gar nicht dargestellt werden. Dies kann allerdings erst im Falle einer Fehlermeldung erfolgen. Auch muss davon ausgegangen werden, dass keine vollständige Überprüfung stattfindet, so dass von einer Korrektheit der Übersetzung insgesamt nicht ausgegangen werden kann.

Aufgrund dieser Überlegungen muss die Überbrückung der Unterschiede zwischen den Domänen dezidiert betrachtet werden. Deshalb lässt sich das GCM auf folgende Sicht (Abbildung 16) einschränken, wobei die verschiedenen Domänen in den Vordergrund treten:

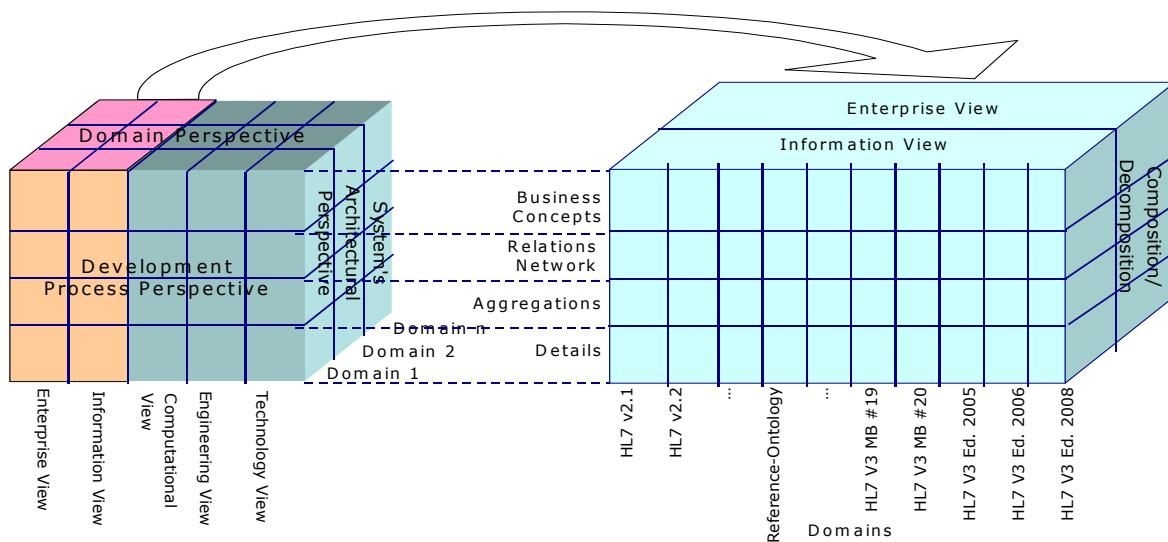


Abbildung 16: Reduktion des GCM auf den Enterprise View [Blob2002, Blob2007b]

Damit lassen sich prinzipiell beliebig viele Domänen parallel betrachten. Eine direkte Abbildung jeder Domäne auf jede andere erscheint aber weder sinnvoll noch handhabbar. Deshalb tritt die Überlegung, einen Mediator zu nutzen, wieder in den Vordergrund. Für den hier betrachteten Anwendungsfall muss dies eine Domäne sein, die der eigentlichen Anwendung entspricht – oder zumindest relativ nahe kommt, um notwendige Anpassungen (Korrekturen und Ergänzungen) möglichst gering ausfallen zu lassen.

## 7. Wissen und Wissensrepräsentation

Die Hauptaufgabe dieser Arbeit ist die Etablierung semantischer Interoperabilität zwischen an sich inkompatiblen Kommunikationsstandards. Dazu ist aber Wissen über die sich hinter den Daten verbergenden Konzepte notwendig, so dass der zentrale Bestandteil der gesamten Arbeit durch die explizite Hinterlegung eben diesen Wissens gebildet wird.

Deshalb wird in diesem Abschnitt erörtert, was Wissen ist und wie es repräsentiert werden kann. Dabei wird die Beschäftigung mit Ontologien auf das Maß beschränkt, welches für die Integration heterogener Informationen erforderlich ist.

## 7.1. *Ontologie*

In der Informatik man versteht unter einer Ontologie eine Wissensrepräsentation eines formal definierten Systems von Begriffen und Relationen. Zusätzlich können Ontologien Inferenz- und Integritätsregeln enthalten, das heißt, Regeln zur Schlussfolgerung und zur Gewährleistung ihrer Gültigkeit. Dabei stellt die Konzeptualisierung eine vereinfachte Sicht auf die Welt dar, die repräsentiert werden soll [Guar1998]. Die Spezifikation der Konzepte muss einer formalen Sprache genügen, um eine brauchbare Repräsentationsgrundlage bilden zu können, die z.B. die Auswertung mit einem Computerprogramm ermöglicht. So kann das spezifizierte Wissen genutzt werden, um neues Wissen abzuleiten bzw. Inkonsistenzen aufzudecken.

Ceusters fordert sogar eine konkrete systematische Beziehung zwischen der realen Welt und der gewählten Repräsentation, d.h. dass es für die Konzepte der Ontologie auch eine Entsprechung in der realen Welt gibt [Ceus2004, SolChu2009].

Für Blobel [Blob2006] "ist eine Ontologie die Summe aller Konzepte", was nach seiner systemorientierten und architekturbasierten Definition implizit auch Relationen umfasst. (Die anderen Autoren definieren das explizit.) Dabei geht er in einem Nachsatz über eine Indirektion mit Hilfe der Definition von "Wissen" darauf ein, dass dann auch Regeln ("production rules, if-then-rules") sowie Frames dazu gehören.

"An ontology is the concrete form of a conceptualisation of a community's knowledge of a domain" [SteGob2000]. Somit kann das generische Komponentenmodell (GCM), auf das diese Arbeit aufbaut, ebenfalls zur Repräsentation von Ontologien benutzt werden – es stellt quasi eine "generische Ontologie" dar.

Wenn eine Ontologie in OWL ausgedrückt wird, dann gilt die Transitivität für Ableitungen in einer Achse, die dann eine is-a Hierarchie ausdrücken. Aber auch für andere Relationen kann Transitivität gelten [Lutz2008]:

$$\mathcal{T} \models C \sqsubseteq D \text{ und } \mathcal{T} \models D \sqsubseteq E \text{ impliziert } \mathcal{T} \models C \sqsubseteq E$$

Das gilt nicht unbedingt für die Reflexivität, die wie folgt definiert ist [Lutz2008]:

$$\mathcal{T} \models C \sqsubseteq C$$

Allerdings gibt es meistens die Umkehrfunktion, die eine vereinfachte Handhabung erlaubt. Von diesem Sachverhalt [Hell2006] wird später beim Aufbau und Generieren der Communication Standards Ontology (CSO) Gebrauch gemacht:

$$\text{hasXXXX} \equiv \text{isXXXXof}^{-1}$$



$$\forall x \forall y (\text{hasXXXX}(x,y) \leftrightarrow \text{isXXXXof}(y,x))$$

Die Entwicklung von Ontologien erfolgt nach gewissen Regeln, die im weiteren Verlauf der Arbeit erläutert werden [RDF, OWL], wobei bestimmte Repräsentationsmittel (z.B. Proposition Logics, Description Logics) stärkere Einschränkungen im Vergleich zu den generischen Definitionen mit sich bringen können. Darüber hinaus kann für einen bestimmten Anwendungszweck nicht entschieden werden, wie eine Ontologie aussehen muss. Es bestehen immer mehrere Möglichkeiten einer geeigneten Repräsentation [NoyMcG, Guar1998, Harr2004].

### 7.1.1. Einsatz von Agenten

Der Erfolg oder Misserfolg einer Ontologie sollte daran gemessen werden, inwieweit die Agenten (beispielsweise Reasoner<sup>10</sup>) operational mit Daten/Wissen umgehen können [Harr2004]. Dieser Aspekt wird im Ausblick zu dieser Arbeit näher untersucht. Zuerst muss eine für den Anwendungszweck ausreichende Konzeptualisierung in Form einer Wissensrepräsentation in den Vordergrund gestellt werden.

Harris hat dazu 5 Ebenen deklariert:

Ebene	Inhalt
0	<b>Ontology Definition Language:</b> Eine Ontologie muss in einer formalen Sprache ausgedrückt werden. Hier bieten sich RDF [RDF] und OWL [OWL] an.
1	<b>Data Structures:</b> Alle Ontologien benötigen Datenstrukturen. In Abhängigkeit von der gewählten formalen Sprache können diese als Tabellen, Slots, Statements oder anders ausfallen. Welche Sprache auch gewählt wird, die Datenstruktur muss zum Anwendungszweck passen.
2	<b>Assertions and Constraints:</b> Neben den Datenstrukturen kann eine Ontologie Annahmen und Einschränkungen enthalten, die als Regeln formuliert sind und sich als Beziehung zwischen den Datenelementen auswirken. Hierbei kann zwischen Integritäts- und Inferenzregeln unterschieden werden. Erstere definieren die Gültigkeitsbedingungen zwischen den einzelnen Datenelementen und werden als Upper-Level-Ontologien bereitgestellt, letztere erzeugen neue Schlussfolgerungen auf Basis der bereits vorhandenen Daten.
3	<b>Reference Data:</b> Viele Ontologien verweisen auf Referenzdaten in Form von Vokabularen, Taxonomien und Thesauren. Diese werden als Komponenten oder Klassifikatoren benötigt.
4	<b>Operational Data:</b> Eine Ontologie bildet eine Struktur, in der Instanzinformationen aus der realen Welt abgebildet werden.

<sup>10</sup> Ein Reasoner überprüft die Konsistenz einer Ontologie.

Die Unterscheidung in verschiedene Ebenen hat Auswirkungen auf die Abbildung der Informationen in einer Ontologie. In anderen Arbeiten [Brachmann, Levesque 1985] wird dies T-Box (terminological component/ Ebene 1+2+3) und A-Box (assertional Component/ Ebene 4) genannt.

Damit neben dem Reasoning intelligente Agenten ihre Arbeit verrichten können und eine Implementierung vereinfacht wird, sollte eine Hierarchiebildung sowohl bei den Konzepten („is-a“) als auch deren (explizit definierten) Relationen vorgenommen werden.

### 7.1.2. Hierarchien/Typisierung

Ontologien können in verschiedene Klassen eingeteilt werden, die wiederum hierarchisch angeordnet werden können. Guarino (Abbildung 17) und Blobel (Abbildung 18) gehen hier unterschiedlich vor:

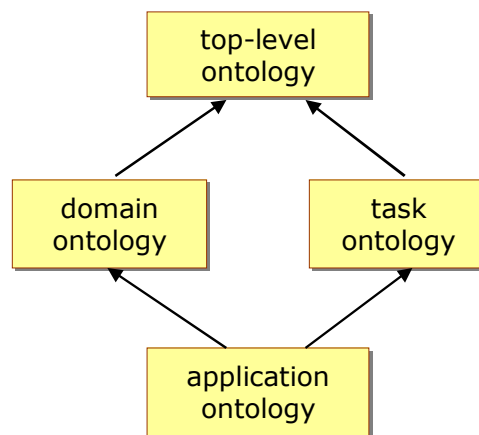


Abbildung 17: Ontologie-Typen [Guar1997, Guar1998]

Eine Top-Level-Ontologie führt als Basis-Ontologie die grundlegenden Konzepte (beispielsweise Zeit, Aktionen, Ereignisse) ein, die unabhängig von einer konkreten Fragestellung sind und die dann von weiteren Ontologien aufgegriffen und verfeinert werden können. So dient eine Domänen-Ontologie der Wissensrepräsentation von Fachinformationen (beispielsweise Diagnosen), während eine Task-Ontologie sich mit der Repräsentation von Aufgaben beschäftigt (beispielsweise die Feststellung einer Diagnose). Eine Anwendungsontologie führt dann das vorhandene Fachwissen wieder mit den Arbeitsabläufen (Workflows) zusammen [Guar1998].

Nach Gruber/Westenthaler [GruWes2006] ist es sehr unwahrscheinlich, dass es eine einzige Ontologie – Anm.: die beiden beziehen sich hier auf Anwendungsontologien - geben wird, die alle Aufgaben abdeckt. Aus der philosophischen Betrachtung heraus ist diese Vermutung ebenso sicher, da eine Abbildung der Realität immer aus einer subjektiven Perspektive heraus und vor dem Hintergrund des Anwendungszwecks erfolgt. Auch das GCM leitet aus seinem System- und Architekturansatz diese quasi

Unbeschränktheit ab. Daher werden Anwendungen vielmehr auf eine Hierarchie von Ontologien (s.u.) zurückgreifen, die z.T. unterschiedlich stark/gut formalisiert sind.

Westenthaler [West2005] klassifiziert Ontologien weiter anhand ihres Skopus (Abstraktion des Modells), der Akzeptanz des Modells sowie der Ausdrucksfähigkeit (Anforderungen an die Repräsentationssprache).

Es zählt lediglich, dass die Ontologie an dem Einsatzzweck orientiert aufgebaut ist [GruWes2006, Guar1998]. Dabei kann von der "Wahrheit" abstrahiert werden, um die Ziele des Einsatzzwecks zu erreichen [Grub2003]. Dies widerspricht der Sichtweise von B. Smith und stellt somit eine unüberbrückbare Divergenz dar [SmiCeu2006].

Das Modell und seine Komponenten (Konzepte) richten sich aber nach dem Ziel und den Intentionen des Modellierers. Außerdem wird zwischen den Komponentenkonzepten, den Aggregationskonzepten und den Policykonzepten unterschieden, was den Kombinationen aus den Referenzen entspricht. Im GCM wird jede Domäne durch eine oder mehrere Ontologien repräsentiert [Blob2006].

Damit ist aber die erstgenannte Hierarchisierung nach [Guar1997] äußerst fraglich. Der Inhalt einer "Task-Ontologie" ist bspw. im Gegensatz zu einer "Domain Ontology" unklar, weil zur Ausführung von "Tasks", so wie sie in einer Anwendung vorkommen, der Bezug zur Domänenontologie notwendig ist. Daher bietet sich eine streng hierarchische Darstellung nach dem Generischen Komponentenmodell an:

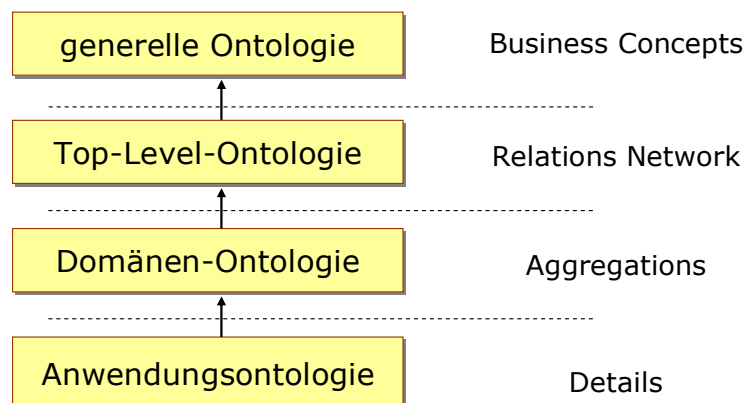


Abbildung 18: Ontologie-Typen [Blob2009b]

Je nach Ausgestaltung der Domänen ergeben sich hier ebenfalls unterschiedliche, an dem Einsatzzweck ausgerichtete Ontologien, die in Einklang zu [GruWes2006] stehen.

Abschließend ist nochmals zu betonen, dass im GCM Ontologiesysteme beschrieben und entwickelt werden (Abbildung 9) als auch entwickelte Ontologien für die Präsentation von Domänenwissen im Kontext von domänenspezifischen Systemen (Deployment of Ontologies) benutzt werden können.

### 7.1.3. Repräsentationsformen

Ontologien lassen sich unterschiedlich darstellen: Wie in Abbildung 9 aufgezeigt, fängt das mit einfachen Termlisten und simplen Glossaren an und geht über einfache Thesauren bis hin zur Universallogik.

### 7.1.4. T-Box/A-Box

In einer Ontologie werden die Informationen, wie vorangehend bereits erläutert, in zwei verschiedenen *Boxen*<sup>11</sup> abgelegt: In der einen Box, der T-Box, werden die Mengen – sprich die Klassen (Konzepte), Relationen und die dazugehörigen Restriktionen – formalisiert, in der anderen, der A-Box, werden die Referenzen zu den Instanzen (Individuen) abgelegt, die die Klassen instantiiieren.

### 7.1.5. Nutzung des Wissens

Um eine möglichst hohe Auswertbarkeit der Informationen zu bekommen sollte möglichst viel des vorhandenen Wissens explizit ausformuliert werden:

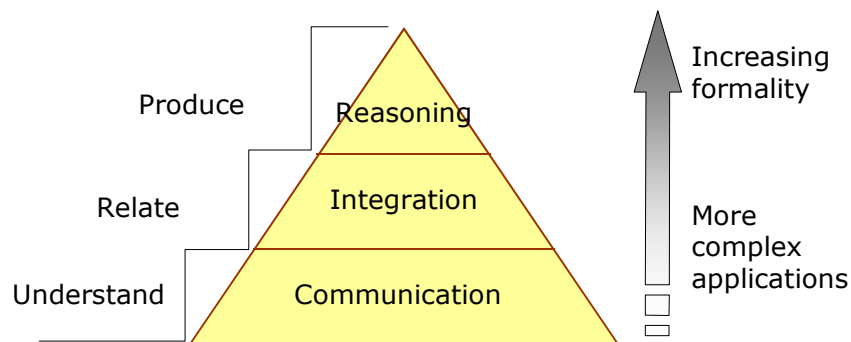


Abbildung 19: Mikas Pyramide der zunehmenden Formalisierung [MiAk2004]

Verschiedene Zwecke der Verwendung von Wissen stellen unterschiedliche Anforderungen an die Wissensbereitstellung. Je komplexer die Anforderungen an die Anwendungen von der Kommunikation von Wissen über die Wissensintegration bis zur Wissensverarbeitung sind, umso weiterentwickelt müssen die maschinelle Verarbeitbarkeit von Wissen und damit der Formalisierungsgrad sein (siehe Abbildung 19). Andere Autoren unterteilen den Zweck von Wissensmodellen in 3 andere Kategorien, die das Verständnis ebenfalls erleichtern [TeVeGe]:

1. Kommunikation
2. Interoperabilität
3. System-Engineering: Spezifikation, Verlässlichkeit und Wiederverwendbarkeit

---

<sup>11</sup> Diese Konvention der Benennung hat sich nicht bewährt und wird deshalb nicht weiter verfolgt. Da sie in "älteren" Arbeiten verwendet wird, wird sie hier zumindest prinzipiell erläutert.

### 7.1.6. First-Order-Logik Definition

Ontologien können auch nach der Repräsentation der Aussagen klassifiziert werden. Da diese Aussagen auf logischen Kalkülen basieren, bieten die verschiedenen Logik-Systeme die Grundlage dazu.

Mittels First-Order-Logik lassen sich Ontologien sinngemäß analog zur Darstellung in [Blob2000] wie folgt definieren:

$$O = (K, R, \pi)$$

Wobei

$K$  : die Menge der Konzepte

und

$R$ : die Menge der Relationen

ist, so dass sich für

$$\pi = \{ r(k_1, k_2): r \in R, k_1, k_2 \in K \}$$

bzw.

$$\pi = \{ f(k_1)=k_2: f \in R, k_1, k_2 \in K \}$$

ergibt.

In der ersten Definition ist zu berücksichtigen, dass die Relation  $r$   $n$ -är sein kann, d.h. auch nur ein einzelnes Konzept verwendet oder sogar mehr als 2 Konzepte zueinander in Beziehung setzt.

Komplexere Konzepte lassen sich auch als Teil- oder Subontologie darstellen:

$$O_{\text{sub}} \subseteq O$$

mit

$$O_{\text{sub}} = (K_{\text{sub}}, R_{\text{sub}}, \pi_{\text{sub}})$$

und

$$K_{\text{sub}} \subseteq K, R_{\text{sub}} \subseteq R, \pi_{\text{sub}} \subseteq \pi$$

wobei

$$\forall k_i, k_j \in K_{\text{sub}}, \forall f: f \in R_{\text{sub}}: f(k_i) = k_j$$

Hierbei ist zu betonen, dass die prinzipielle Betrachtung von Regeln im Kontext der Diskussion von Wissensrepräsentation und Ontologien durch die dort angewandten Repräsentationsmittel wie z.B. die als Grundlage für OWL dienende Description Logics

starke Einschränkungen erfahren kann. Die hier getroffenen mengenbezogenen Statements sind in DL nur für Instanzen, nicht aber für Klassen möglich. Jedoch ist dieses Statement unter dem Gesichtspunkt der rekursiven Natur der Klasse-Instanz-Beziehung in Meta-Sprachen zu betrachten. So kann die Instanz einer Klasse auf einem anderen Granularitätsniveau wiederum als Klasse mit neuer Instanz interpretiert werden, was durch den systemtheoretischen und architekturzentrierten Ansatz des GCM sehr gut demonstriert werden kann. Viele Ansätze, die diesem Paradigma nicht folgen, tun sich mit dieser offenen Interpretation leider sehr schwer.

### **7.1.7. Reasoning**

Ein weiterer Aspekt in der Nutzung von Ontologien ist die Konsistenz: Ontologien werden im Prinzip aus einfachen Statements aufgebaut, aus denen sich Schlussfolgerungen ziehen lassen. Dies führt zu Problemen, wenn die Statements Widersprüche enthalten. Die Widerspruchsfreiheit wird durch sogenannte Reasoner (wie beispielsweise RACER, Pellet, Hermit) überprüft.

## **7.2. *Ontologie- und KR-Sprachen***

Es stellt sich die Frage nach einer Formalisierung von Ontologien, so dass die domänen-spezifischen Informationen in einer computerverwertbaren Form hinterlegt sind.

### **7.2.1. Prolog und LISP**

Die klassischen Sprachen der künstlichen Intelligenz sind (waren) Prolog – „Programming in Logic“ – und LISP – „List Processing“ – [ShoMe1984, Lisp, Nils1982]. Hiermit lassen sich (regel- oder framebasierte) Anwendungen (beispielsweise [OemCre1991]) programmieren, um bestimmte domänenspezifische Fragestellungen gezielt zu lösen.

So wurde die erste Version (MYCIN) des in dieser Arbeit verwendeten Werkzeugs Protégé<sup>12</sup> mit LISP erstellt [GenMus2003, Shor1976].

### **7.2.2. Resource Description Framework (RDF)**

In den letzten Jahren hat die Verbreitung und Nutzung des Internets immens zugenommen. Die Anzahl der verfügbaren Webseiten ist unüberschaubar groß, Inhalte über bloße Stichworte nicht mehr aufzufinden und Mehrdeutigkeiten auf syntaktischer Ebene nicht mehr aufzulösen. Tim Berners-Lee hat deshalb in einem Grundsatzartikel zum Internet die Frage aufgeworfen, wie effizienter mit den Inhalten auf den einzelnen Webseiten umgegangen werden kann [Bern2001]. Darüber hat sich eine Beschreibungssprache – das Resource Description Framework (RDF) [RDF] – entwickelt, mit deren Hilfe Webseiten annotiert werden können/sollen.

---

<sup>12</sup> Die neueren Version 3 und 4 sind in Java programmiert. Die Version 4.x benutzt sogar den OSGi Plug-in Mechanismus.

In diesem Zusammenhang ist der "AAA-Slogan" entstanden: „Anyone can say anything about any topic“ [AllHen2008], ein vielfach brauchbarer, aber im Kontext standardisierter Lösungen problematischer Ansatz.

### 7.2.3. Web Ontology Language (OWL)

Viele der aktuellen Arbeiten im Bereich des Semantic Web setzen auf die Web Ontology Language (OWL, [OWL]), eine Sprache, die eine Erweiterung des RDF darstellt.

OWL bietet die Möglichkeit, Informationen/Wissen in Form von Klassen zu modellieren und darzustellen, die dann über DatatypeProperties zusätzliche Eigenschaften haben können. Darüber hinaus besteht die Möglichkeit, Relationen zwischen Klassen (Object-Property) aufzubauen.

Eine weitere grundlegende Eigenschaft von OWL ist die Möglichkeit, Instanzen zu den Klassen zu bilden.

Ein Reasoner (Fact++, Pellet) hat dann die Möglichkeit, die Konsistenz der Ontologie zu verifizieren.

Die für die Wissensrepräsentation erarbeiteten Begriffe/Konzepte werden in OWL als Klassen repräsentiert. In dieser Arbeit wird daher Klasse und Konzept synonym verwendet, wenn es um den Umgang mit OWL geht.

Dabei wird hier ein anderes Problem ersichtlich: Wie wird gewährleistet, dass zwei verschiedene Personen bei Darstellung in OWL unter einem Begriff dasselbe verstehen, oder andersherum, dass dieselbe Bezeichnung nicht zwei verschiedene Konzepte repräsentiert? Um dieses Problem anzugehen, wurden innerhalb von OWL Namensräume eingeführt. Jeder Namensraum – die verwendete Terminologie – stellt mit den eingesetzten/definierten Relationen, die auch wieder Konzepte sind, eine eigenständige Ontologie dar. Somit müssen die einzelnen Klassen/Konzepte aus den verschiedenen Namensräumen bei Gleichheit explizit miteinander in Beziehung gesetzt werden.

Im Anhang wird dazu noch näher auf die Grundlagen und die verschiedenen Dialekte (Version 1.0 und 2.0) von OWL eingegangen.

## 8. Ordnungsstrukturen für Wissensrepräsentation

Die vorangehend erläuterten Definitionen sollen in diesem Abschnitt inhaltlich weiter aufbereitet werden:

### 8.1. Beobachtungseinheiten

Eine Beobachtungseinheit ist definiert als "die kleinste Einheit in einem Beobachtungsverfahren, an der Messungen (Beobachtungen) vorgenommen werden können". Hierbei erfolgt die Festlegung einer Beobachtungseinheit sowohl in zeitlicher als auch inhaltlicher Hinsicht. [Grum1974]

Beobachtungseinheiten werden durch Merkmale und deren Relationen repräsentiert. Je nach Kontext kann eine Beobachtungseinheit unterschiedlich repräsentiert werden: [Blob1987]

Tabelle 5: Domänenspezifische Beobachtungseinheiten [Blob1987]

Bereich	Bezeichnung	Information
Medizin-Informatik	Merkmal	Ausprägung
Mathematik	Variable	Wert
Statistik	Zufallsvariable	mögliches Ergebnis, Ereignis

Die Merkmale im Bereich der Medizin-Informatik können weiter unterteilt werden:

- qualitative Merkmale (Ausprägungen: Begriffe)
  - nominale Merkmale (ohne natürliche Ordnung)
  - ordinale Merkmale (geordnet)
- quantitative Merkmale (Ausprägungen: Zahlen)
  - stetige Merkmale (kontinuierliche Größe)
  - diskrete Merkmale (abzählbar)

## 8.2. Begriffsmengen

Bestimmte Klassen von Beobachtungen werden demnach durch Begriffsmengen repräsentiert. Begriffe sind somit Ausprägungen der Merkmale spezieller Beobachtungseinheiten, die eine komplexe Struktur annehmen können. Ein Begriff fasst Eigenschaften eines Objektes der (realen) bzw. der Modell-Welt zusammen (vgl. [Inge2004]).

Das der Klassifikation von Begriffsmengen zugrunde liegende Modell kann ebenfalls komplex sein. Hierbei besteht ein solches Modell neben den Merkmalen und deren Ausprägungen (semantischen Konzepten, Begriffen), aus den Relationen zwischen den Begriffen und der Klassifikation zur expliziten Darstellung der Relation [Blob1987],

### 8.2.1. Ordnungsstrukturen

Die vorab genannten Beziehungen haben u.a. zum Ziel, eine Ordnung in der Menge der Konzepte/Begriffe zu etablieren. Die nachfolgend erläuterten Ordnungssysteme erlauben mitunter verschiedene Ordnungsstrukturen, die über Beziehungen der eingetragenen Informationen/Konzepte hergestellt werden. Eine is-a-Relation erstellt eine Beziehung zwischen Instanzen und dem zugrunde liegenden Konzept (Klasse) (Mein BMW ist ein Auto.) sowie zwischen Konzepten (Klassen) als Spezialisierung (Ein Auto ist ein Fahrzeug.)

Eine andere Relation ist eine Komponentenrelation (part-of), so dass ein Objekt Teil eines anderen ist. (Der Daumen ist Teil der Hand.) (DIN 2342, [Zaiß]). Darüber hinaus kann eine part-of Relation durchaus komplizierterer Natur sein: [SchuKum2006, SmiCeU2005]



Bei ausschließlicher Nutzung der einfachen is-a-Beziehung in einer eindimensionalen Ordnungsstruktur ergibt sich somit eine einfache Baumstruktur. Bei gleichzeitiger Nutzung mehrerer Relationstypen über dieselbe Konzeptmenge lassen sich mitunter komplexe Gebilde definieren, deren Administration damit ebenfalls zunehmend schwieriger wird. Snomed CT [Snomed CT] mit seinen 18+3 Achsen ist ein gutes Beispiel hierfür. LOINC [LOINC] hat im Gegenzug nur 6 Achsen, welche allerdings nur implizit über eine externe Beschreibung vorhanden sind.

Diese bisher genannten Ordnungsstrukturen stützen sich auf die Semantik der Konzepte. Alternativ dazu gibt es auch syntaktische Ordnungsstrukturen, wie es beispielsweise für das Alphabet definiert ist ("b" kommt nach "a").

### 8.2.2. Vokabular

"Ein Vokabular enthält nur Begriffe, die durch eine kontrollierende Instanz (z.B. einen Administrator innerhalb einer Einrichtung, eine nationale Organisationseinheit usw.) aufgenommen wurden. Es dient der Vereinheitlichung der Sprache/ Dokumentation zur Verbesserung des personen- und einrichtungsübergreifenden Verständnisses für dokumentierte Sachverhalte." [Haas2006]

**vo-cab-u-lary** (Merriam Webster,[Merriam b])

"a list or collection of words or of words and phrases usually alphabetically arranged and explained or defined"

$$V = \{ \text{"Menge aller Codes"} \}$$

$$V = \{ \text{"Menge aller Terme"} \}$$

Nach obigen Definitionen handelt es sich beim Vokabular um eine Menge von Begriffen. Cimino benutzt den Term "controlled vocabulary" um die Kodierung narrativen Texts zu unterstützen [Cimi1998]. Der Tatbestand "controlled" soll dabei aussagen, dass nicht beliebige Terme benutzt werden sollen/dürfen, sondern eine gewisse Aufwärtskompatibilität eingehalten werden soll. Dies wird auch in [Zaiß] mit dem "regulierenden Charakter" betont.

Peter Haas [Haas2005] benutzt die Duden-Definition von Wernke [Wern2001], wonach es sich beim Vokabular um "einen Wortschatz handelt, dessen man sich bedient".

Damit bleibt offen, ob es sich bei einem Vokabular um Codes oder natürlichsprachliche Zeichen wie Wörter oder Mehrwort-Terme handelt.

### 8.2.3. Thesaurus

"Ein Thesaurus im Bereich der Information und Dokumentation ist eine geordnete Zusammenstellung von Begriffen (und nicht Konzepten) und ihrer (vorwiegend natürlich-

sprachlichen) Bezeichnung, die in einem Dokumentationsgebiet zum Indexieren, Speichern und Wiederauffinden dient." [DIN 1463]

Zaiß [Zaiß] legt den Schwerpunkt auf "die Zusammenfassung von Wörtern mit gleicher Bedeutung über Deskriptoren". Dabei sind (Poly)Hierarchien über Taxonomien und Partonomien ebenfalls zulässig.

#### **8.2.4. Terminologie**

Terminologie: Systematik eines Fachwortschatzes [Duden]

Formale Terminologien ergänzen den Informationsinhalt unter Zuhilfenahme von formalen Repräsentationsformalisten wie beispielsweise Deskriptionslogiken, um die Beziehung zwischen den enthaltenen Informationen für Schlussfolgerungen (Deduktionen) nutzen zu können. [Zaiß]

$$T = \{ c_i : \forall i : \exists s : \text{Repräsentation}(c_i) = s \}$$

Zur Sicherung von Konsistenzen und somit zur Verbesserung der „Interoperabilität“ von Standards wurde durch das ISO TC 215 „Health Informatics“ ein tool-basiertes Glossar [ISO 28379] aller existierenden Spezifikationen entwickelt. Für die Erarbeitung neuer Standards muss auf dieses Glossar zurückgegriffen werden. Neue Begriffe dürfen nur über ein entsprechendes Gremium eingepflegt werden.

IHE hat das gleiche Problem: Die diversen Technical Frameworks benutzen bestimmte Terme, die aus den zugrunde liegenden Standards entstammen. Dadurch ergibt sich zwangsläufig, dass derselbe Term mit unterschiedlichen Bedeutungen verwendet wird. [IHE Vol.0].

#### **8.2.5. Nomenklatur**

"Nomenklaturen sind Mengen möglichst eindeutiger Benennungen (Termini) der Begriffe eines Fachgebietes." [Zaiß]

Zur Herstellung der Eindeutigkeit einer Benennung ist ein Regelwerk erforderlich, dass sich an der Fachdomäne orientiert.

Durch dieses Regelwerk werden Beziehungen festgelegt, die eine Ordnungsstruktur aufbauen. Allerdings basiert diese nicht auf semantischen, sondern lediglich auf syntaktischen Konzepten.

#### **8.2.6. Taxonomie**

Eine Taxonomie nutzt zum Aufbau des Ordnungssystems nur is-a-Relationen [Zaiß]. Auf diese Weise werden Klassen von allgemeineren Objekten weiter spezialisiert. Dies geschieht sowohl durch Hinzufügen weiterer neuer Eigenschaften als auch durch Einschränkung der bereits vorhandenen.

### **8.2.7. Partonomie**

Eine Partonomie hingegen setzt zum Aufbau des Ordnungssystems nur part-of-Relationen ein [Zaiß]. Damit werden der Aufbau von Objekten aus Teilen beschrieben.

### **8.2.8. Regeln**

Eine Möglichkeit der Hinterlegung von Wissen ist das explizite Ausformulieren aller Tatsachen. Dies führt aber zu einer Datenmenge, die nicht mehr bewältigt werden kann. Es ist daher einfacher, insbesondere Aktionen und Konsistenzbedingungen über sog. „wenn-dann“-Relationen auszudrücken.

### **8.2.9. Frames**

Eine weitere Möglichkeit Wissen auszudrücken ist in Form von Stereotypen oder Frames. Dabei werden Informationen oder Wissen in gleichartigen Strukturen (Slots) abgelegt, so dass ein entsprechender Interpreter, wenn er über das Wissen über die in diesen Strukturen abgelegten Informationen verfügt, agieren kann. Die zum Verständnis notwendige formale Information ist somit implizit in den Frames enthalten und nicht explizit mit formaler Logik ausgedrückt, was nicht ausschließt, dass Slots logische Statements enthalten können. [Maida]

## ***8.3. UML-Klassendiagramm der Begriffsmengen***

Die o.g. Begriffe sind, wie aus der jeweiligen Definition hervorgeht, nur schwer auseinander zu halten. Es bietet sich hier an, die einzelnen Begriffsmengen über ein UML-Klassendiagramm<sup>13</sup> in Beziehung zu setzen, das die Unterschiede herausarbeitet:

---

<sup>13</sup> UML stellt verschiedene Diagrammtypen zur Verfügung, um Modelle formal zu beschreiben.

Grundsätzlich ist es möglich, die UML-Diagramme auch in Form einer Ontologie auszudrücken. Dies wird später anhand der Scores demonstriert.

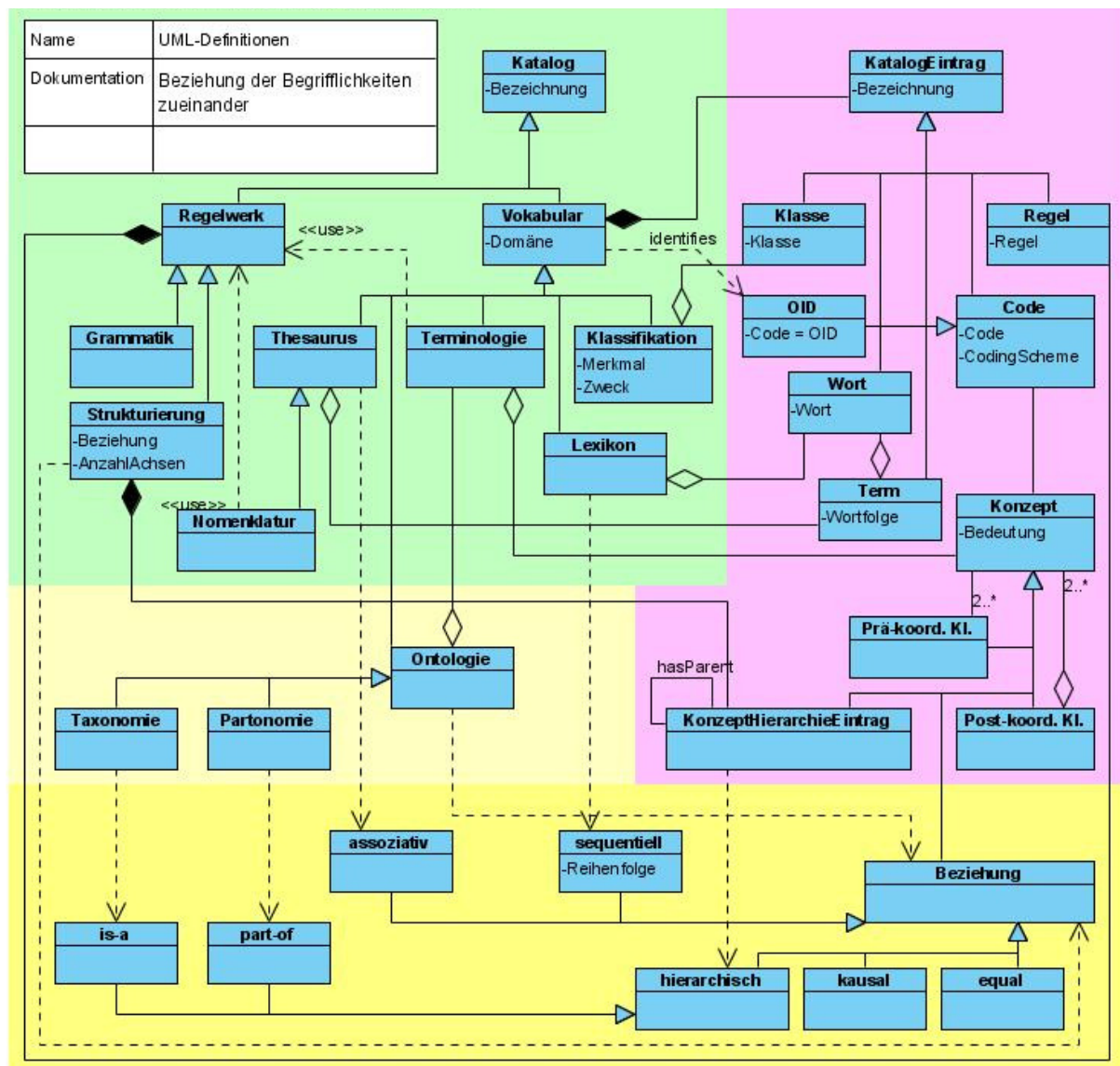


Abbildung 20: UML-Klassendiagramm der Begriffsmengen

Die im UML-Klassendiagramm [OemBlo2010a] angedeuteten, aber nicht weiter ausgeführten assoziativen Relationen lassen sich noch weiter spezialisieren, wie die nachfolgende Grafik zeigt. Für diese Arbeit ist das jedoch nicht relevant, so dass im Nachfolgenden darauf nicht weiter Bezug genommen wird.

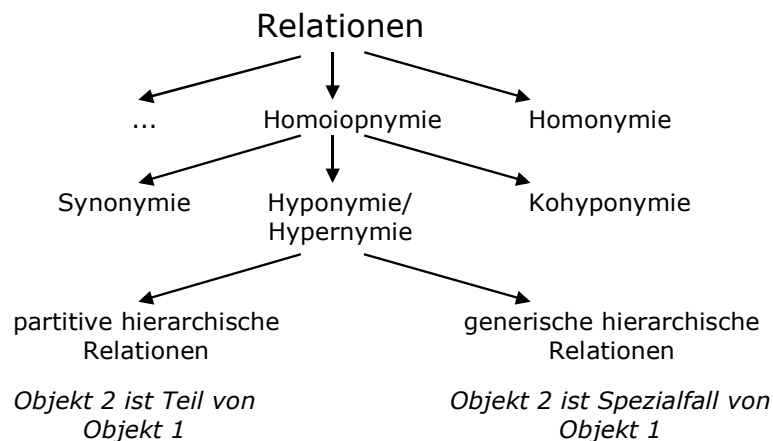


Abbildung 21: Spezialisierungen der assoziativen Relation [Blob1987]

Im Wesentlichen handelt es sich bei den vorgenannten Begriffsmengen um Kataloge, die unterschiedliche Arten von Einträgen (Worte, Wortgruppen, Codes, Regeln) in unterschiedlicher Art und Weise (mit oder ohne Hierarchie) zueinander in Beziehung (einzeln/mehrfach) setzen. Wie die Details exakt ausgedrückt werden, soll nicht Bestandteil dieser Grafik sein: So ist ein Vokabular ein Katalog, der lediglich Einträge enthält. Ein Regelwerk hingegen ist ein Katalog, bei dessen Einträgen es sich um Regeln handelt. (An dieser Stelle soll nicht darauf eingegangen werden, wie diese Regeln aufgebaut und formuliert sind.) Eine Nomenklatur wiederum ist ein Thesaurus, dessen Einträge nach einem bestimmten Regelwerk aufgebaut sind. Ein Thesaurus ist ein Vokabular, das Terme (Begriffe/Wortfolgen) über Assoziationen (Akronyme, Synonyme) in Beziehung zueinander bringt. Eine Terminologie benutzt im Gegensatz zu einem Thesaurus Konzepte.

Eine Taxonomie bzw. Partonomie benutzt zwar is-a bzw. part-of Relationen, allerdings nur innerhalb einer Domäne, da die in Beziehung gesetzten Konzepte dasselbe Bezugssystem haben müssen.

Durch die Aufweitung der möglichen Relationen unter Einbindung der Gleichheitsrelation kann auch eine Brücke zu anderen Domänen und damit anderen Konzepten gebaut werden.

#### **8.4. Einordnung in das GCM**

Die beschriebenen Begrifflichkeiten lassen sich im GCM einordnen. Die Hintergrundfarben in vorhergehender Grafik markieren die Klassen, die in 4 Kategorien auf 3 Ebenen unterteilt werden können. So sind die Katalogeinträge in rot markiert. Diese stellen Basis-konzepte dar, die im GCM in der unteren Ebene zu finden sind.

Die Kataloge selber repräsentieren bereits Funktionen, da sie die Katalogeinträge je nach Semantik benutzen. Diese sind in grün markiert und auf der zweiten Ebene angesiedelt. Ontologien (hellgelb) – und damit auch Taxonomien und Partonomien – setzen andere Konzepte in vielfältiger Art und Weise zueinander in Beziehung. (Die verschiedenen Möglichkeiten für Relationen sind in gelb dargestellt.) Dies kann dann auf einer Ebene

innerhalb einer Domäne geschehen, aber auch über verschiedene Ebenen einer Domäne oder über verschiedene Domänen hinweg.

Business Objekte werden nicht durch diese Begrifflichkeiten abgedeckt. Daher bleibt diese Ebene leer.

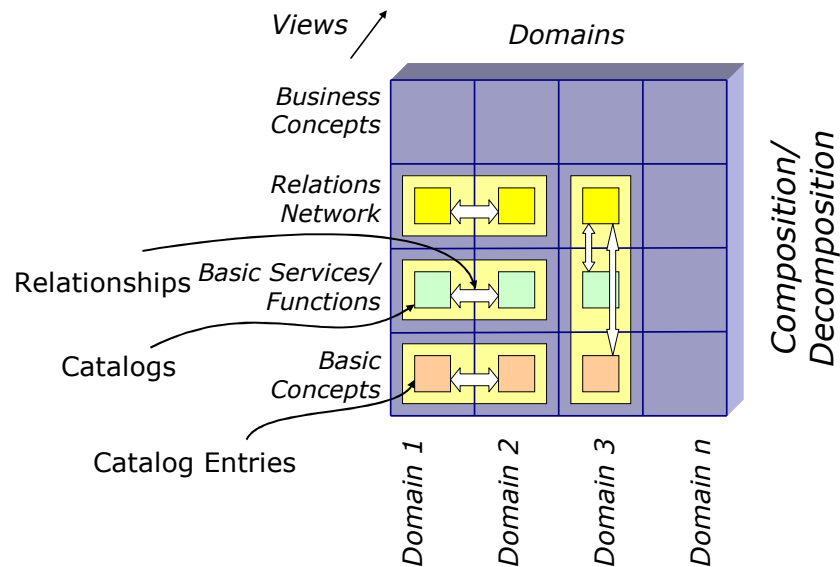


Abbildung 22: Einordnung in das GCM (1)

In gleicher Weise können Ontologien genutzt werden, um auch die verschiedenen Sichten zu überbrücken. Insgesamt wird die Funktionalität von Beziehungen, die über Ontologien ausgedrückt werden, genutzt, um in dem dreidimensionalen Raum des GCM die einzelnen Konzepte zu verbinden:

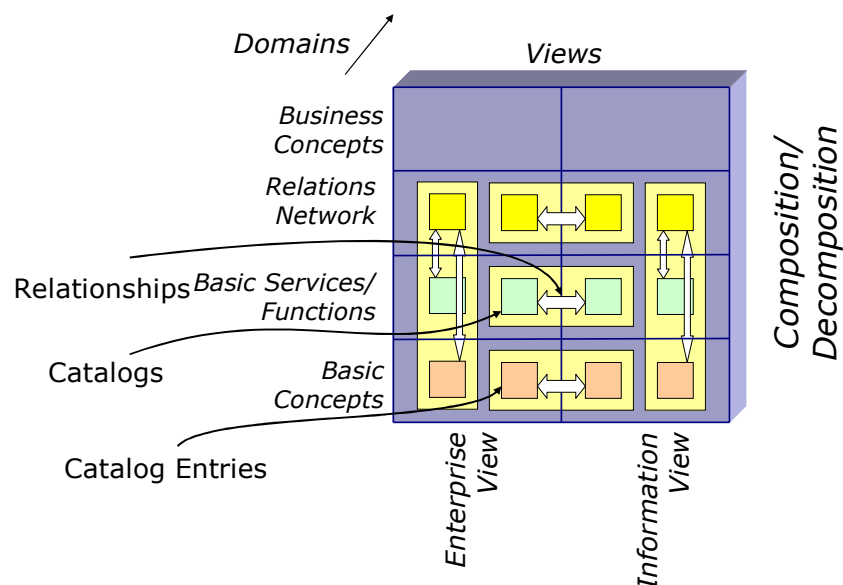


Abbildung 23: Einordnung in das GCM (2)

Zusammenfassend kann an dieser Stelle festgehalten werden, dass das Ziel dieser Arbeit, eine semantische Austauschbarkeit über verschiedene Kommunikationsstandards

hinweg zu ermöglichen, somit nur in einem Architektur-Framework mit Hilfe von Ontologien realisiert werden kann, da nur diese die notwendigen Ausdrucksmittel zur Verfügung stellen.

## D. Ergebnisse

Nachdem in dem vorhergehenden Kapitel dargelegt worden ist, dass eine Verbindung verschiedener Domänen – in die die beiden Kommunikationsstandards HL7 v2.x und V3 zweifelsohne fallen – nur durch den Einsatz von Ontologien hergestellt werden kann, müssen diese Domänen soweit formalisiert werden, dass daraus die notwendigen Konzepte und die dazugehörigen Relationen abgeleitet werden können. Dies wird in den nachfolgenden Kapiteln vorgenommen.

## 9. Analyse der Kommunikationsstandards

HL7 [HL7] hat im Rahmen der Entwicklungsarbeiten an den Standards keine Ontologie definiert. Die Version 2.x wurde sogar ohne eine einheitliche Philosophie für die Erstellung der Segmente und Nachrichten entwickelt. Diese Schwachstelle suchte man mit der Entwicklung der Version 3 über das Message Developing Framework (MDF, [HL7-MDF]) und später das HL7 Development Framework (HDF, [HL7-HDF]) zu eliminieren. Aber auch hier wurde keine Ontologie explizit definiert. Das HDF benutzt ein inzwischen normatives Referenz-Informations-Modell (RIM) zur Absicherung der Konsistenz bei der Entwicklung der Domänenmodelle (D-MIMs). Mit der Entwicklung einer HL7-Rahmenarchitektur (bei HL7 heißt diese SAEAF [SAEAF]), die dem GCM folgen soll und damit ontologiebasiert sein muss, wird dieses Manko überwunden werden und der Übergang vom Nachrichtenparadigma zum Architekturparadigma vollzogen. Bis dahin wird aber noch eine gewisse Zeit vergehen. Außerdem werden die bisher entstandenen Probleme dadurch nicht gelöst.

Grundlage dieser Arbeit sind als erster Schritt deshalb UML-Klassendiagramme [UML], die für die entsprechenden Kommunikationsstandards entwickelt wurden. Diese werden in diesem Kapitel zunächst im Detail erläutert, bevor Anmerkungen dazu gemacht werden, um die Schwachstellen für eine Umsetzung in eine Ontologie zu identifizieren.

Im Anschlusskapitel werden dann erste Überlegungen angestellt, wie daraus eine Referenz-Ontologie entwickelt werden kann, welche die verschiedenen Kommunikationsstandards umfasst und ein Mapping erlaubt.



### **9.1. UML-Modell für HL7 v2.x**

In der HL7-Spezifikation der v2.x-Familie existiert weder ein explizites Konzept-Modell noch ein Daten-Modell, das Auskunft über den Aufbau des Standards gibt. Über die Reverse-Engineering-Methode lässt sich ein derartiges Modell [OemDud1996, OemSto2000] ableiten und das darüber herausgearbeitete Wissen in Form eines UML-Klassendiagramms darstellen. (Darauf basiert zum größten Teil auch die HL7-Comprehensive-Database – zumindest was die Speicherung der relevanten Daten für den Standard angeht.) Dieses Diagramm wird später zur Ableitung der Darstellung in OWL benötigt.

Das nachfolgende Modell integriert alle relevanten Aspekte und orientiert sich inhaltlich an der Zielstellung dieser Arbeit. Die in den Modellen angegebenen Buchstaben dienen zur Referenz auf Kommentare, die nach den anschließenden Erläuterungen aufgeführt werden:

# Entwicklung einer ontologiebasierten Architektur zur Sicherung semantischer Interoperabilität zwischen Kommunikationsstandards im Gesundheitswesen

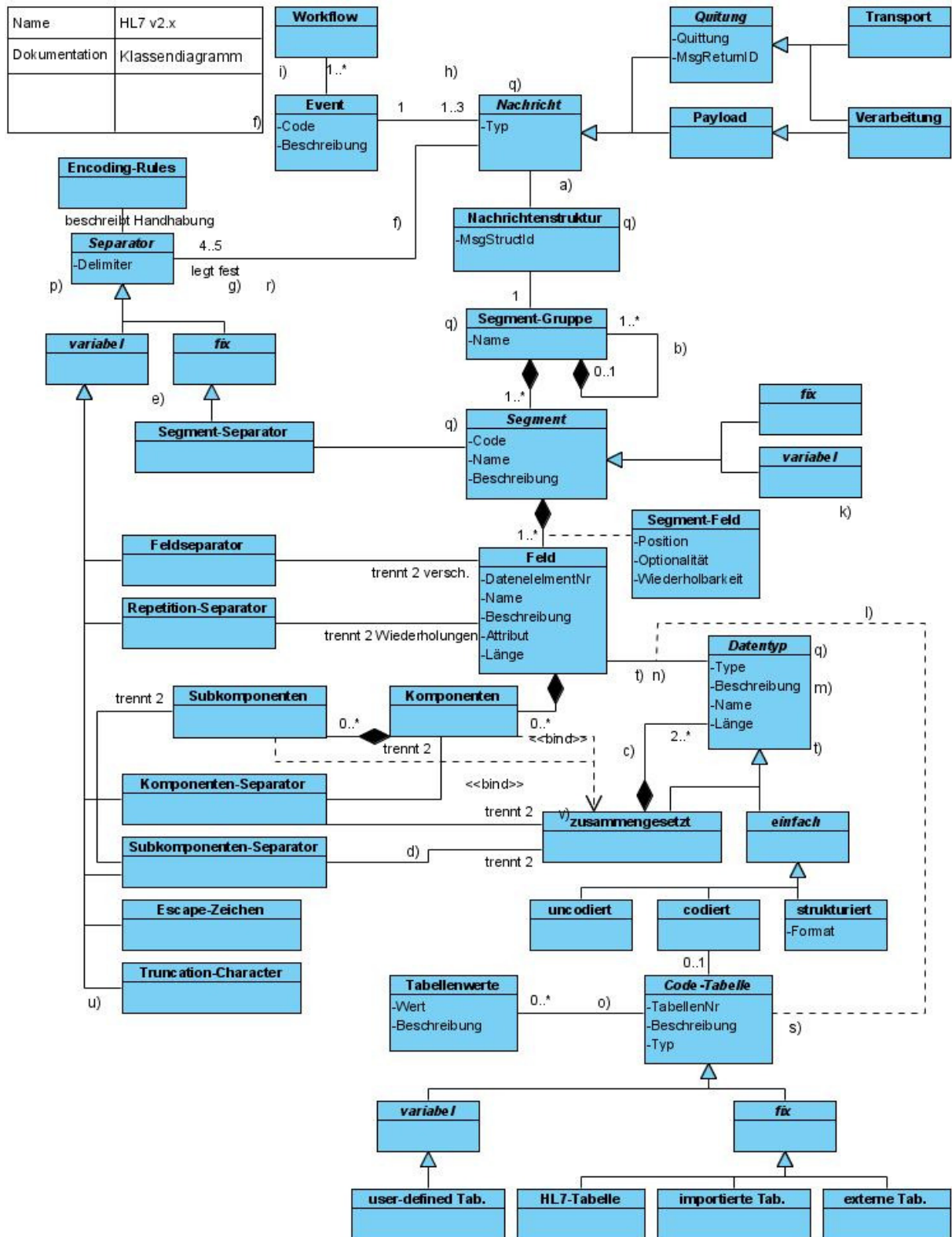


Abbildung 24: UML-HL7 v2.x Modell

Ein Ereignis (in der realen Welt) löst eine Nachricht aus, die wiederum von ein oder zwei Quittungsnachrichten bestätigt werden kann. Ob eine und welche Quittungsnachricht ausgelöst wird, wird im Nachrichtenkopf (Messenger Header - MSH) angegeben, der als erstes Segment in der Nachricht angegeben ist. Im Nachrichtenstandard selbst werden zu jedem Event die direkt ausgelöste Nachricht sowie die (Transport)-Quittung ange-

geben. Für das einfache Nachrichtenszenario ist das ausreichend. Es sind jedoch vereinzelt Nachrichten spezifiziert, für die das nicht ausreicht. Für einen sinnvollen Ablauf sind die Nachrichten zu einem "Workflow" zu kombinieren. Dies gilt beispielsweise für die Order-Entry-Nachrichten. Allerdings ist das nur implizit als textuelle Information im Standard vorhanden und muss durch den Anwender herausgelesen werden.

Jede Nachricht besteht aus einer Reihe von Segmenten in einer bestimmten Anordnung, die über die abstrakte Nachrichtenstruktur (Abstract Message Specification, AMS) definiert wird. Das Konzept der Nachrichtenstruktur und einer dazugehörigen Identifikation wurde erst in der Version 2.3.1 ("Migration" in nachfolgender Abbildung) auf der Basis eines Vorschlags des Autors eingeführt (rechte Seite in nachfolgender Grafik). Davor wurde den Ereignissen direkt eine Segmentreihenfolge zugeordnet (linke Seite). Teilweise wurden einem Ereignis sogar mehrere Nachrichten (unbeabsichtigt) zugeordnet, die ohne einen eindeutigen Identifikator nicht zu verwalten waren und somit zu Irritationen und falschen Implementierungen geführt haben.

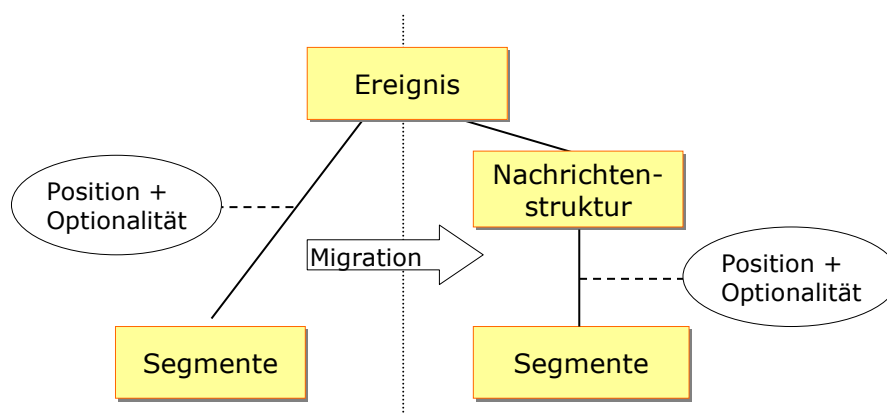


Abbildung 25: Vereinfachte schematische Zuordnung der Nachrichtenstruktur

Genau genommen muss jeder Nachricht eine Nachrichtenstruktur zugeordnet werden, so dass jedes Ereignis in genau eine Nachricht(enstruktur) mündet. Diese Nachrichtenstruktur ist demnach im Prinzip einer Segment-Gruppe gleichzusetzen, die als äußere Klammer die komplette Nachricht umfasst. Daraus resultiert dann die spätere OWL-Modellierung einer Nachricht als Spezialfall einer Nachrichtengruppe.

Eine Segment-Gruppe wiederum enthält eine Reihe von Segmenten bzw. Segmentgruppen. Hierüber ist eine beliebig tiefe Schachtelung möglich.

Jedes Segment enthält wiederum eine Reihe von Feldern.

Jedes Feld hat einen Datentyp, der entweder aus einer oder mehreren Komponenten besteht. Über das Mehrkomponentenfeld entsteht eine Rekursion auf den Datentyp, die

über die Anzahl der Delimiter/Trennsymbole (im ER7-Encoding<sup>14</sup>) auf zwei Ebenen begrenzt<sup>15</sup> wird.

Auf der anderen Seite sorgt der Datentyp für die syntaktische Repräsentation der Information. Zusätzlich werden hierüber Tabellen angebunden. Auf der anderen Seite werden semantische Zusammenhänge in der Kombination mit den Komponenten und Subkomponenten der Felder realisiert.

Das Einzelkomponentenfeld kann in uncodierte/unstrukturierte, strukturierte oder sogar codierte Felder unterteilt werden. Die erste Variante kann als String beliebige Daten aufnehmen. Für den strukturierten Datentyp gibt es zumindest eine Formatvorgabe, wie dies beispielsweise für Zeitangaben der Fall ist. Im Falle eines codierten Datentyps ist sogar eine Code-Tabelle hinterlegt, die bestimmte Werte vorschreibt. Diese Code-Tabellen können unterteilt werden in solche mit festen Wertvorgaben oder diejenigen, für die die Werte erst noch festgeschrieben werden müssen (user-defined tables).

Eine Besonderheit in diesem Modell nehmen die sog. Trenner/Separatoren/Delimiter ein. Diese sind für die Umsetzung nach den Encodingregeln notwendig. Zur Sicherstellung einer HL7-konformen Realisierung ist die Implementierung in der hier vorgestellten Form notwendig - auch wenn dies in einigen nationalen Projekten nicht als wichtig erachtet wurde [HL7-DE 2007]. Für diese Arbeit ist dieser Umstand allerdings ohne Belang.

Einige Datentypen sollen für eine nähere Betrachtung explizit benannt werden. Dies würde das vorhergehende allgemeine Diagramm überfrachten, so dass dies nachfolgend an ausgewählten Beispielen noch einmal explizit dargestellt ist.

---

<sup>14</sup> Encoding Rules 7 – Darstellung der Daten im HL7-v2.x-Standardformat mit den "|" - Zeichen (vgl. Kap.2 von v2.x)

<sup>15</sup> Diese Beschränkung könnte durch weitere Trennsymbole aufgehoben werden.

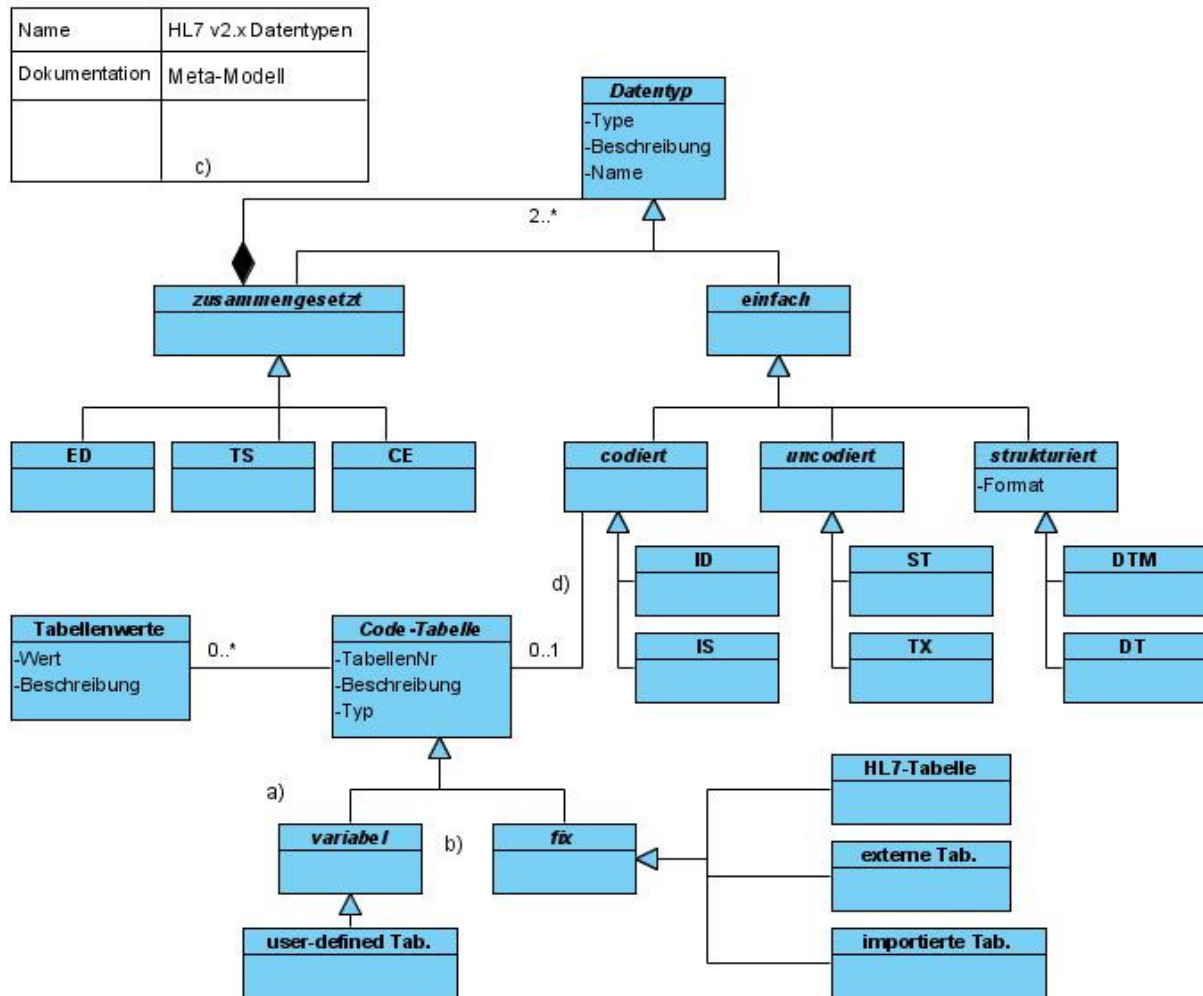


Abbildung 26: UML-HL7 v2.x Datentypen Modell

Bevor das Modell für HL7-V3 vorgestellt wird, werden die Anmerkungen und Kritikpunkte, die durch Buchstaben in den beiden Diagrammen angedeutet sind, ausführlich erläutert/aufgeführt.

### 9.1.1. Anmerkungen zum HL7-v2.x-Modell

Das erste Modell (Abbildung 24) beschreibt lediglich die Komponenten des HL7-v2.x-Modells mit ihren statischen Beziehungen entsprechend den Basiskonzepten, infolge des Fehlens einer Ontologie (der Summe der Konzepte und aller möglichen Relationen) nicht aber die Konsequenzen variabler Komponenten und Relationen. Diese können aus dem Standard auch nur schwer herausgelesen werden. Dies ist eine Ursache für die vielfach vorkommende falsche Implementierung durch Hersteller. Diese Aspekte sollen nun kurz erläutert werden, auch um die spätere Abbildung in OWL zu erleichtern. Die nachfolgend aufgelisteten Punkte beziehen sich deshalb auf die markierten Stellen im UML-Klassendiagramm:

- a) Die Abbildung einer Nachricht auf eine Nachrichtenstruktur wird nicht immer korrekt vorgenommen. Zum Einen gibt es Ereignisse, zu denen mehrere Nach-

richtenstrukturen existieren (dies kommt z.Bsp. in älteren Versionen im Bereich Order Entry vor), zum Anderen sind dieselben Nachrichtenstrukturen nicht immer exakt gleich definiert. Beides wird grundsätzlich durch den verwendeten Texteditor und die historisch bedingte Struktur des Standards in der textuellen Repräsentation verursacht.

Dies würde sich im Prinzip über Nachrichtenstrukturen lösen lassen, die nur an einer Stelle im Standard hinterlegt sind (wie weiter vorn schon aufgeführt, ist seit v.2.3.1 auf Betreiben des Autors dieser Arbeit ein Identifikator für Nachrichtenstrukturen im Standard enthalten). Es ist allerdings zu beobachten, dass sich dieses Problem in der neuesten Version 2.6 verschärft hat. Erschwert wird die Gewährleistung der Konsistenz durch einen früheren Beschluss des ARB (Architecture Review Board von HL7 Inc.), nicht mehr zu verwendende Strukturen nicht weiter zu pflegen.

- b) Die Rekursion von Segmentgruppen erlaubt grundsätzlich beliebig tiefe Schachtelungen, die ein Parser korrekt identifizieren muss, d.h. eine Segmentgruppe kann eine andere enthalten. So enthält die Gruppe „ORDER“ Details („ORDER\_DETAILS“), die wiederum auf vorhergehende Ergebnisse („OBSERVATION“) zurückgreift.

In einer Nachricht gemäß ER7 haben diese Gruppen jedoch keine mitübermittelte Bezeichnung.

Auch ist die Bezeichnung nicht in allen Versionen von 2.x vorhanden, da sie erst mit v2.5 aufgeführt wurde. Über die HL7-Datenbank sind die Bezeichnungen aber in alle 2er Versionen hinunter migriert worden, so dass ein XML-Encoding für alle Versionen prinzipiell möglich ist.

Des Weiteren ist festzuhalten, dass nicht alle Nachrichtenstrukturen für einen Parser eindeutig definiert sind. Hier wäre beispielsweise die Nachricht OML\_O33 zu nennen, die mehrdeutige Interpretationen zulässt.

- c) Die Zusammensetzung von Mehrkomponentenfeldern über weitere Datentypen erlaubt im Prinzip beliebig tiefe Schachtelungen, da jede Komponente einen Datentyp hat, der wiederum mehrere Komponenten umfassen kann. Es gibt aber nur Separatoren für 3 Ebenen: Felder ohne Komponenten, Komponenten und Subkomponenten.

Die HL7-Arbeitsgruppe InM muss das Auftreten tiefergehender Schachtelungen verhindern.

- d) Welcher Separator (Feld, Komponente, Subkomponente, ..) der richtige ist, kann im Diagramm nicht festgelegt werden, da durch die rekursive Darstellung keine individuelle Beziehung definiert werden kann.

Aus einer weitergehenden formalisierten ontologischen Betrachtung wäre dies aber möglich.

- e) Die Unterteilung in fixe (fest definierte) und variable (in Nachrichteninstanzen individuell veränderbare) Separatoren erfordert einen leicht erhöhten Aufwand bei

---

der Implementierung. Dass Separatoren variabel sein können, wurde aber nicht in allen Implementierungen berücksichtigt. Ohne eine Flexibilität durch Verwendung alternativer Separatoren ist Testen/Zertifizieren [OemBlo2007b] von Nachrichten hinsichtlich korrekten Encodings nur mit überhöhtem Aufwand möglich.

- f) Die meisten Implementierungen gehen von fixen Separatoren aus.  
In den deutschen Nachrichtenprofilen [HL7-DE 2007] sind die Separatoren über Restriktionen soweit eingeschränkt, dass sie nicht variiert werden können.
- g) Es müssen mindestens 4 der 5 variablen Separatoren festgelegt werden, aber nicht unbedingt alle 5. Dies kann dazu führen, dass zum Einen bei bestimmten Implementierungen auf falsche Nachrichtenstrukturen zugegriffen wird und damit Informationen abgeleitet werden, die so nicht vorhanden sind. Zum Anderen kann ein standardmäßig genutzter Delimiter im normalen Zeichenvorrat genutzt werden. Ab v2.7 werden alle Separatoren verbindlich.  
Mit dem fixen Separator gibt es insgesamt 6 Trenner.
- h) Für die max. drei Nachrichtenstrukturen werden festgelegt:
- die Nachricht mit Inhalt (Payload) als Initiator, sowie
  - eine Transport-Quittung und
  - eine Verarbeitungs-Quittung als Antwort.
- Damit werden die logischen Relationen festgelegt. In konkreten realen Situationen mit Broadcast-Nachrichten an alle beteiligten Systeme können somit fast beliebig viele Antworten auf eine initiale Nachricht eintreffen.
- i) Der Workflow ist nicht explizit im Standard festgelegt. Mitunter werden die logischen Antworten über ein anderes Event übermittelt und somit nicht erkannt. Ein höherwertiger Workflow könnte sie erkennbar machen.  
IHE ("Integrating the Healthcare Enterprise", [IHE]) hat sich zum Ziel gesetzt, Workflows zu definieren und dafür spezialisierte Nachrichtenprofile zu definieren. Allerdings erfolgt dies nicht anhand einer strukturierten Vorgehensweise.
- k) Variable Segmente zeichnen sich dadurch aus, dass das letzte Feld als Solches beliebig oft vorkommen kann. Dies ist beispielsweise beim QPD-Segment der Fall, wo das letzte dritte Feld auch als viertes, fünftes etc. vorkommen kann, um individuelle Parameter zu definieren.  
Nach den HL7-v2.xml-Vorgaben [HL7 v2.xml] lassen sich dafür keine adäquaten XML-Schema definieren.
- l) Mitunter wird einem zusammengesetzten Feld eine Tabelle zugeordnet. Damit ist gemeint, dass die Tabelle einer bestimmten Komponente des Datentyps zuzuordnen ist. Genau genommen müssten die Datentypen hierfür spezialisiert werden, beispielsweise CWE („*coded with exceptions*“) oder CNE („*coded no exceptions*“).
- m) Für eine korrekte Abbildung der Datentypen in einer dem Standard kompatiblen Art und Weise muss für die älteren HL7-Versionen (v2.1 und v2.2) eine Zwischenschicht eingezogen werden. Ursache dafür ist ein generischer Datentyp (CM), der zum damaligen Zeitpunkt nicht im Detail aufgelöst wurde.

- Über die HL7-Datenbank wurde dieses Problem gelöst und in den Nachfolgeversionen berücksichtigt [OemDud1996].
- n) Ein Datentyp ist entweder einfach oder zusammengesetzt.  
Dasselbe gilt für ein Feld.  
Im Prinzip ist das eine doppelte Abbildung desselben Sachverhaltes. Zur korrekten Zuordnung der Semantik müssen Inhalte von Einzelkomponenten dem komplexen Produkt zugeordnet werden. Über den Datentyp geht dieser Zusammenhang verloren.
- o) Code-Tabellen mit festen vorgegebenen Werten haben mit [n..m] eine von [0..\*] abweichende Kardinalität.
- p) Die einzelnen Trennzeichen (Separatoren) sind im Prinzip Tabellenwerte. Dieser Sachverhalt spielt in diesem Modell aber keine Rolle.
- q) Die in obiger Grafik genannten Klassen stellen im Prinzip Spezialisierungen von HL7-Tabellen dar, d.h., diese werden z.T. bei HL7 in Kapitel 2C als individuelle Tabellen, allerdings ohne spezifische Attribute, aufgelistet. Im Standard wird diesem Sachverhalt allerdings keine Rechnung getragen, so dass die Informationen dupliziert werden und damit zu Inkonsistenzen führen, die nur mühsam per Hand in den Griff zu bekommen sind.  
Hiervon sind insbesondere die Tabellen 0003 (Events), 0076 (Message Types) und 0354 (Message Structure IDs) betroffen.  
Letztere führen insofern noch zu zusätzlichen Problemen, weil Nachrichten mehrfach an unterschiedlichen Stellen gepflegt werden und dadurch Inkonsistenzen nicht zu vermeiden sind. Eine Zuordnung einer Beschreibung bzw. eines Zweckes würde die Nutzung vereinfachen.
- r) Im Standard sind nur Default-Werte für die variablen Trennzeichen angegeben. Allerdings setzen fast alle Implementierungen diese fest-kodiert um, so dass eine Anpassung nicht möglich ist.  
Indirekt führt dies zu einem nicht HL7-konformen Encoding.  
Als Beispiel sei hierzu angeführt, dass viele Implementierungen den Nachrichten-anfang mit Hilfe von "MSH|" anstelle von "<0x13>MSH" testen.
- s) Der Object Identifier (OID) ist der Tabelle und nicht dem semantischen Inhalt ("Vocabulary Domain") zugeordnet.  
Damit kann derzeit nicht geprüft werden, ob diese Tabellen inhaltlich (semantisch) identisch sind.  
Beispiel: Von Version 2.3.1 zu 2.4 wurde die Tabelle 0001 (Sex) um weitere Codes ("A" und "N") ergänzt. Damit ändert sich die Bedeutung des Codes "U" (unknown), da mit den neuen Codes bei "unbekannt" jetzt weiter differenziert werden kann. Deshalb müsste die Tabelle 0001 in der Version 2.4 einen neuen Objekt-ID (OID) bekommen.



- t) Derzeit (bis einschließlich Version 2.6) ist die Länge definiert als die Maximallänge<sup>16</sup>, die ein Wert einnehmen kann. Für eine HL7-konforme Implementierung bedeutet das, dass dies die exakte Länge ist, die von der Schnittstelle unterstützt werden muss. Allerdings passen die Hersteller ihre Längen (in der Datenbank) nicht entsprechend an. Bei Abweichungen – egal ob nach oben oder unten – führt dies zu Problemen bei der Übermittlung von Daten.
- Für die HL7-Version 2.7 ist deshalb beabsichtigt, dieses Problem auf Vorschlag<sup>17</sup> des Autors durch "Schließen der Hintertür" und normative Maximallängen zu beseitigen: Es sollen nur Längen festgelegt werden, bei denen man sich auf normative Werte einigen kann. Darüber hinaus kann es Empfehlungen für sinnvolle Implementierungen geben. Diese Empfehlungen sollen durch eine spezielle Notation kenntlich gemacht werden.
- Des Weiteren soll die Verantwortlichkeit für Längenangaben in die Konformanzprofile verlagert werden. In "Implementable Profiles" wird eine Längenangabe dann zur Pflicht.
- Zusätzlich soll die Längenangabe weiter in eine Minimal- und Maximallänge präzisiert werden. In Kombination mit der Übertragungsrichtung (Sender oder Empfänger) und den dazugehörigen Konformanzregeln lässt sich damit präzise testen, ob zwei Schnittstellen zueinander in Bezug auf die Länge kompatibel sind.
- Für nicht-normative Längenvorgaben wurde ein alternativer Mechanismus entwickelt: Hier sollen zu unterstützende Minimallängen definiert werden. Mit diesen Weiterentwicklungen wird der status quo des bisherigen Standards einerseits erhalten, andererseits die problematische "Hintertür" geschlossen.
- Zu diesem Vorschlag kommt noch eine Angabe, ob ein Feld abgeschnitten werden darf oder nicht. Damit wird verhindert, dass relevante Informationen wie Codes verfälscht werden können.
- u) Ab Version 2.7 kommt noch ein spezieller Delimiter hinzu, um abgeschnittene Informationen (Truncation) kenntlich zu machen. (Diese Erweiterung geht originär auf einen Vorschlag des Autors zur Verbesserung der Längenbehandlung zurück,

---

<sup>16</sup> Im Standard gibt es noch ein "Hintertürchen", das bei Vorhandensein eines "Site-specific Agreements" für die Maximallänge eine abweichende Angabe zulässt. Der Autor ist der Meinung, dass jede laufende Installation einer Anwendung ein derartiges Übereinkommen darstellt und somit jede Länge zulässt, solange "es funktioniert". Damit lässt sich aber eine spezielle Maximallänge nicht eindeutig festhalten, da bei einer Installation seltenst Längen abgesprochen oder gar getestet werden.

<sup>17</sup> Dieser Vorschlag wurde in den zuständigen TCs (HL7-InM, HL7-IC) kontrovers diskutiert. Nach der Auffassung des Autors wird durch den Vorschlag eine Verbesserung für alle (Hersteller, Anwender) ohne Qualitätsverlust erzielt. Für die HL7-Version 2.7 wurde aber eine abweichende Darstellung ausgehandelt, die aber im Prinzip den Originalgedanken aufgreift und entsprechend umsetzt.

der aber durch G. Grieve abgewandelt wurde. (Vgl. Punkt t.)

Die Deklaration eines gesonderten Zeichens erlaubt hierbei eine rückwärtskompatible, transparente Behandlung abgeschnittener Informationen.

- v) Komponenten und Subkomponenten sind im Prinzip äquivalent zu den Datentypen, d.h. sie werden über die Datentypen und die dazugehörigen Spezifikationen realisiert. Allerdings kann über Datentypen keine direkte Semantik formalisiert werden, da diese unabhängig von den Feldern definiert worden sind.

Für eine Festlegung der Semantik muß somit das Feld mit berücksichtigt werden.

### **9.1.2. Anmerkungen zum HL7-v2.x-Datentypen-Modell**

In dem Modell zu den Datentypen (Abbildung 26) sind ebenfalls einige Anmerkungen enthalten:

- a) Die Klasse Code-Tabellen ist abstrakt, so dass sie in fixe und variable Tabellen spezialisiert werden muss. Variable Tabellen gestatten eine beliebige Festlegung von erlaubten Werten.
- b) Tabellen mit festgelegten Werten können wiederum in drei verschiedene Typen unterteilt werden, je nach dem, woher diese Werte stammen und wie mit diesen innerhalb des Standards umgegangen wird.  
Für die Umsetzung in dieser Arbeit ist allerdings die Tatsache relevant, dass für diese Tabellen Werte existieren, die zu verwenden sind.
- c) Ein zusammengesetzter Datentyp enthält als Komponenten wiederum entsprechende Datentypen. Damit ist die Kardinalität der Komponenten auf ein Minimum von 2 vorherbestimmt.  
Grundsätzlich ist über diese Modellierung eine beliebig tiefe Rekursion möglich. Bedingt durch die Delimiter ist eine Rekursion allerdings auf zwei Ebenen beschränkt.
- d) Den kodierten Datentypen sind Tabellen zugeordnet. Mitunter sind die Tabellen aber nicht a priori festgelegt.

## **9.2. UML-Modell für HL7 V3**

Wegen der zugrunde liegenden modell-basierten Methodologie und Modellhierarchie ist es mit dem UML-Modell für HL7 Version 3 einfacher als für v2.x, alle notwendigen Details abzubilden. So können die 4+2 Basisklassen des RIM quasi direkt übernommen werden. Dazu kommen dann noch einige Spezialklassen für das Messaging oder beispielsweise die sprachlichen Fähigkeiten ("communication").

Die Attribute der jeweiligen Klassen werden separat modelliert. Mit diesen werden dann die Vocabulary Domains sowie die Datentypen verbunden. Damit folgt die Behandlung einem typischen Komponentenmodell, das mit Komponenten aus anderen Domänen statisch und sogar dynamisch verbunden werden kann, und aus dem durch Constraint-Modellierung beliebige Instanzen abgeleitet werden können.

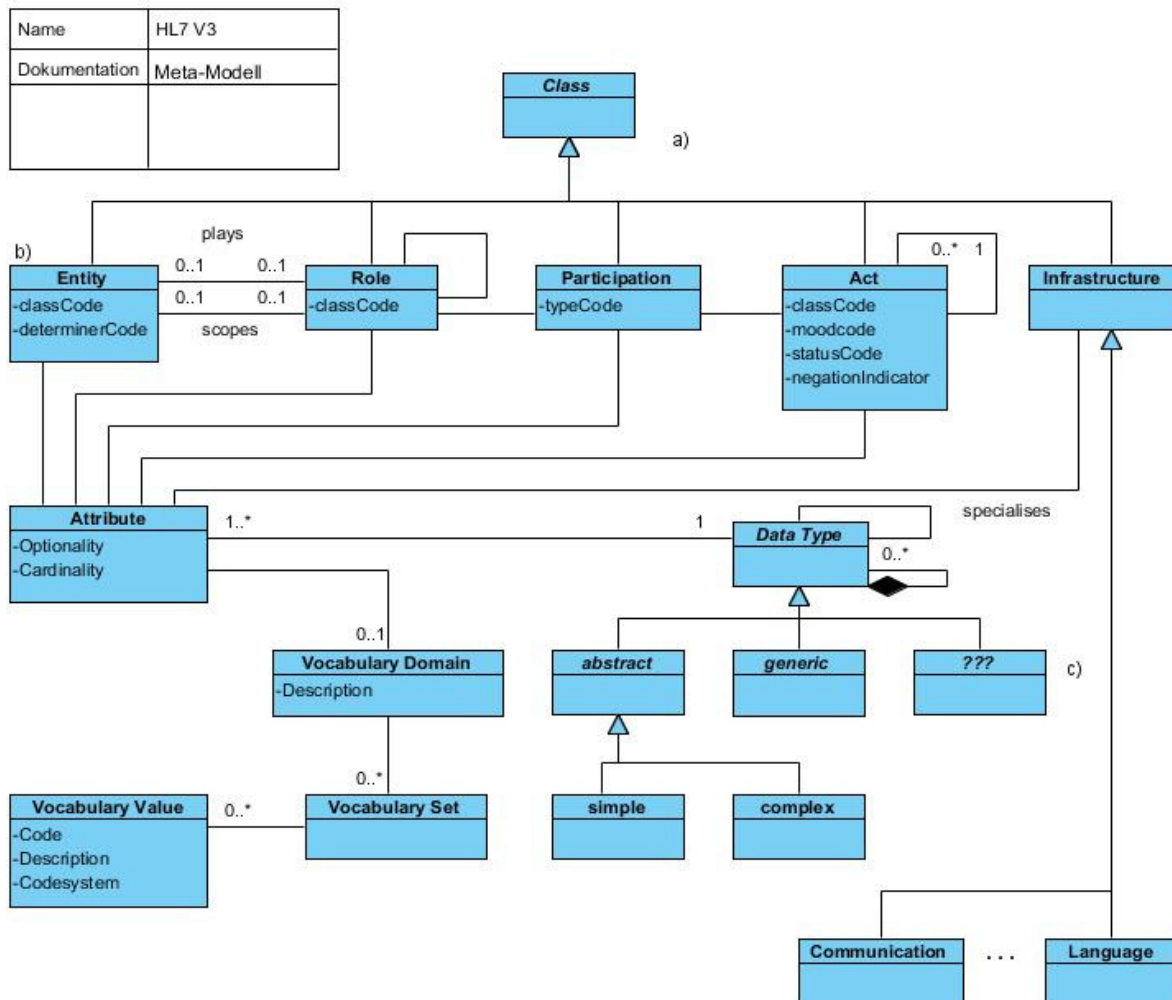


Abbildung 27: UML-HL7 V3 Modell

## Anmerkungen zum HL7-V3-Modell

Bei der HL7-V3-Modellierung ist Folgendes zu beachten:

- Im RIM werden die Klassen nicht von einer gemeinsamen Superklasse abgeleitet. Dies vereinfacht hier nur die Umsetzung mittels OWL. Die diesbezüglichen Vor- oder Nachteile sind nicht Bestandteil dieser Ausarbeitung.
- Die strukturellen Attribute sind den Klassen belassen.
- Hier ist noch zwischen abstrakten Datentypen und deren Realisierung mittels ITS (Implementable Technology Specification) zu unterscheiden.

## 10. Ontologien für Kommunikationsstandards

Bei der Entwicklung von Ontologien für die Systemintegration sind die intendierte Rolle und Architektur der Ontologie, ihre Repräsentation, die intendierten Mapping-Mechanismen sowie die Entwicklung und Pflege der Ontologie (Ontology Engineering) zu definieren, da die genannten Aspekte die resultierende Ontologie wesentlich beeinflussen.

Die Entwicklung von Ontologien für die Kommunikationsstandards geschieht mit OWL/RDF<sup>18</sup> [OWL, RDF] und lässt sich auf die Idee reduzieren, die einzelnen Informationseinheiten in den verschiedenen Darstellungen exakt zu repräsentieren. In einem nachgelagerten Schritt soll mit Hilfe der eingesetzten Technologie eine direkte Abbildung vorgenommen werden.

Komplexere Konstrukte (Mehrkomponentenfelder oder Nachrichtenfragmente) bekommen für die Einheit als Ganzes eindeutige Identifikatoren, die die Definition von Äquivalenzen ermöglichen (vgl. Abbildung 30).

In Analogie zu den auf Seite 44 vorgestellten Ontologietypen lassen sich die in dieser Arbeit zu entwickelnden Ontologien wie folgt in eine Hierarchie einordnen:

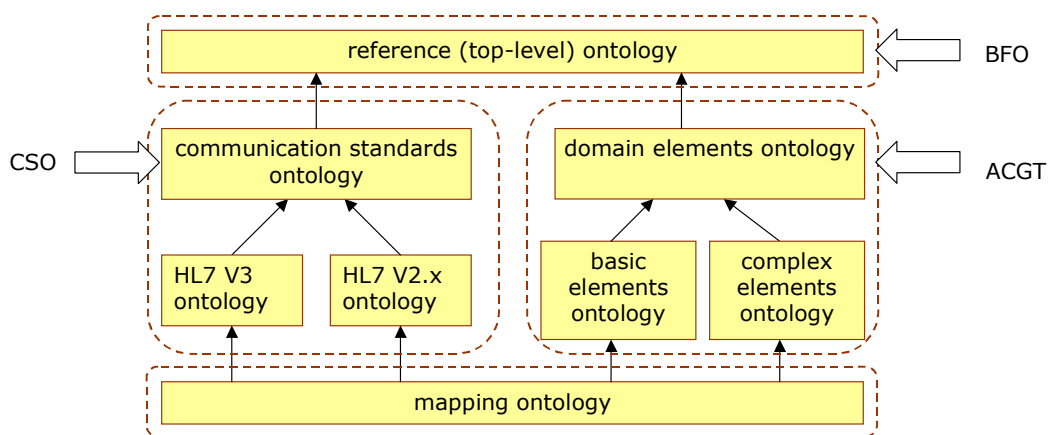


Abbildung 28: Angepasste Ontologie-Typen

Damit sind aus den verschiedenen HL7-Standards jeweils eigene Ontologien abzuleiten. Die Basis bildet eine Grundlagen-Ontologie, die die Basiselemente enthält (s.u.). Dazu kommt dann noch eine domänenspezifische Ontologie ("domain elements"), um komplexere Konstrukte und damit ein Mapping zu realisieren.

Grundsätzlich sind bei diesem Vorgehen folgende Aspekte zu betrachten:

Aspekt:	Erläuterung:
Meta-Modell	Wie sind die einzelnen Klassen in den Meta-Modellen von v2.x und V3 am besten zu repräsentieren?
D-MIM	Domänenmodelle sind die Grundlage für die Nachrichten.
HMD	In HL7 V3 werden zusätzliche Einschränkungen in den HMDs festgelegt. Diese sind für die Repräsentation der Nachricht zu berücksichtigen.
Datentypen	Implementierungstechnisch werden Datentypen zu unterschiedlichen Zwecken eingesetzt. Zum Einen werden diese für eine syntaktische Abbildung benutzt. Wesentlicher ist jedoch die Beschreibung semantischer Zusammenhänge.

<sup>18</sup> Eine einleitende Beschreibung findet sich im Anhang.

---

Vokabular	In HL7 v2.x und V3 werden durch die Methodologie bedingt Informationen über unterschiedlich granulare Vokabularien und unterschiedliche Strukturen abgebildet. Für eine Umsetzung muss dieses Problem gemeistert werden.
Automatismus	Das Ergebnis der Arbeit muss eine möglichst autonome Übersetzung der Informationen ermöglichen. Hierzu wäre die Nutzung von sog. „Reasonern“ für OWL oder automatisch generierte Java-Klassen bzw. XSL-Translations optimal. Ein weiterer Weg zur Automatisierung ist die Generierung der korrespondierenden OWL-Beschreibung aus den gegebenen Modellen.

## **10.1. Grundlagen der Repräsentation medizinischen Wissens**

Bevor eine Übertragung der hinterlegten Informationen vorgenommen werden kann, muss eine geeignete Darstellungsart für das erarbeitete Wissen gefunden werden, die einfach zu handhaben ist. Nachfolgend wird kurz Snomed CT daraufhin untersucht:

### **10.1.1. Relationales Repräsentationsformat von Snomed CT**

Snomed CT [Snomed CT] definiert beispielsweise mit seinen 18+3 Achsen, d.h. Spezialisierungshierarchien, derzeit 67 Relationen, über die Konzepte miteinander in Beziehung gesetzt werden können. Dazu gehören dann auch Funktionen wie „is-a“ oder „hasXXXX“. Eine Beziehungsfunktion kann hierbei ihrerseits wieder als ein eigenes, spezielles Konzept<sup>19</sup> verstanden werden. So werden Objekt-Attribut-Wert-Beziehungen ebenfalls über Beziehungsfunktionen ausgedrückt. Im Falle von Snomed CT können diese Werte relativ leicht wieder als Konzept ausgedrückt werden, was aber nicht in allen Ontologien der Fall ist.

Snomed CT [Snomed, Snomed CT] hat die Informationen auf ein einfaches Textformat reduziert, das prinzipiell aus drei Tabellen besteht. Die weiteren Tabellen für CrossMaps und Subsets spielen in der für diese Arbeit notwendigen Betrachtung keine Rolle, wären aber für ein generisches Tool bedeutsam.

---

<sup>19</sup> Da die Modellkomponente durch ein Konzept (Wissen) beschrieben wird, und auch die Regelwerke Konzepte (Wissen) repräsentieren, wird dieser Sachverhalt nicht als Erweiterung entwickelt, sondern ist GCM-Prinzip.

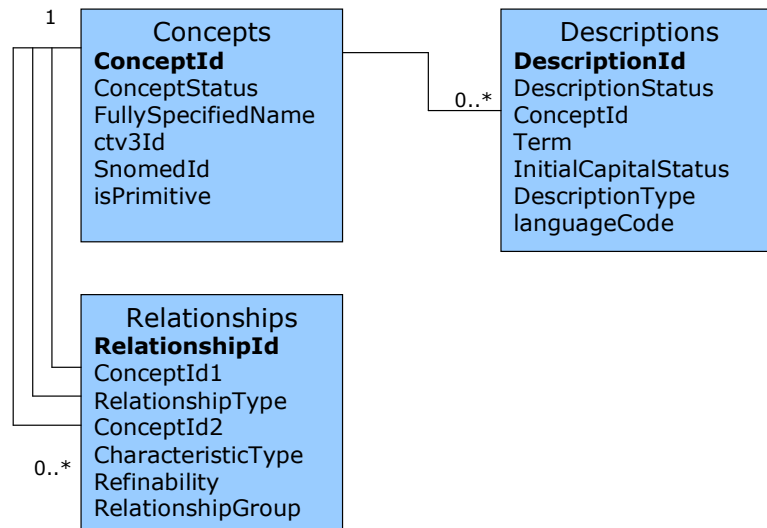


Abbildung 29: Snomed CT – Inhaltsrepräsentation mittels dreier Tabellen

Jedes Konzept wird einzeln dargestellt und über die Tabelle "Concepts" mit einer eigenen ID verwaltet. Einem Konzept können dabei auch mehrere Beschreibungen (Terme, "Descriptions") zugeordnet werden.

Über die Tabelle "Relationships" werden 2 Konzepte miteinander in Beziehung gesetzt. Die Relation ("RelationshipType") ist dabei selbst wiederum ein eigenes Konzept, das über die erste Tabelle verwaltet wird.

### 10.1.2. Relationen und Achsen

Die in der Analyse der Standards vorgestellten Klassen müssten über geeignete Mechanismen auf eine vergleichbare Darstellung reduziert werden. Damit kommt dem Attribut "RelationshipType" eine besondere Rolle zu. Zum Einen sind die Relationen zwischen den Klassen darüber auszudrücken. Hier bieten sich für die Aufgabenstellung in dieser Arbeit ad hoc folgende Konzepte/ObjectProperties<sup>20</sup> an, die später verifiziert und ggf. noch weiter diversifiziert werden:

- is a
- has components (besteht aus)
- is part of
- sequentialize (zur Sortierung von mehreren Komponenten beispielsweise für Listen<sup>21</sup>)
- has datatype
- has value
- use table (Beziehung von Feldern/Komponenten/Subkomponenten zu Tabellen)

---

<sup>20</sup> In anderen Ontologien kommen andere Beziehungen zum Einsatz. So wird in der Gene Ontology (GO) "regulates" verwendet.

<sup>21</sup> In OWL gibt es spezielle Konstrukte, um Listen darzustellen.

---

Zum Anderen müssten die Attribute der Klassen ebenfalls durch eigene Konzepte ausgedrückt werden. Hierfür werden neue Konzepte als zusätzliche Achsen benötigt. Mögliche Attributierungen (Wertausprägungen) wie bei den Längeninformatoren müssen entweder über eigene Konzepte oder OWL-Mechanismen repräsentiert werden:

- Länge
- Optionalität/Usage
- Wiederholungen/Kardinalität

Sowohl bei der Länge (bei v2.x: zunächst nur die Maximallänge, zukünftig auch die Minimallänge sowie die Konformanzlänge) als auch bei der Kardinalität sind prinzipiell beliebige Werte möglich. Eine Abbildung als Konzept ist dabei nicht angebracht.

Bei der Optionalität und Kardinalität ist zu beachten, dass es sich nicht um identische Konzepte, wohl aber um Konzepte mit überlappender Semantik handelt.

### **10.1.3. Konvertierung nach OWL**

Für Snomed CT existiert ein Perl-Script<sup>22</sup>, das den Datenbestand in eine OWL-Repräsentation konvertiert. Ein weiteres Argument für OWL ist die Verfügbarkeit von Tools. Protégé ist ein solches, für das bereits zahlreiche Plug-Ins existieren. Im Folgenden wird deshalb analysiert, welchen strukturellen Aufbau die Daten haben und wie diese nach OWL konvertiert werden können.

## **10.2. *Ontologie-Struktur***

Zur Umsetzung der in der Analyse erarbeiteten Modelle wird eine Ontologie-Struktur benötigt.

### **10.2.1. Überlegungen**

Wie bereits einleitend erläutert erscheint folgende Unterteilung für eine Ontologie-Struktur sinnvoll (Abbildung 30):

---

<sup>22</sup> Dieses Script wurde von Kent Speckman entwickelt, das er auf Nachfrage zur Verfügung stellt.

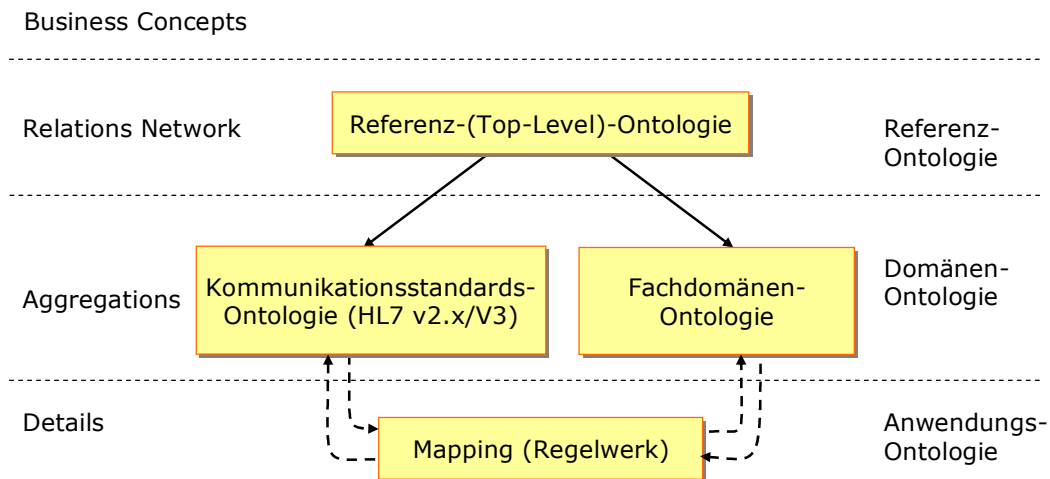


Abbildung 30: Aufteilung der Ontologielevels (für die hier geforderte Architektur)

Anhand des GCM können die verschiedenen HL7-Versionen als Spezialisierung bzw. Verfeinerung einer Referenz-Ontologie als die erste Domäne aufgefasst werden. Wie die vorhergehende Abbildung andeutet, können die beiden Familien von Kommunikationsstandards in zwei parallele Gruppen (v2.x und V3) aufgeteilt werden, die intern derselben Grundlage folgen und über ähnliche Maßnahmen erzeugt werden. Damit könnten sie nach den Prinzipien des GCM direkt miteinander in Beziehung gesetzt werden, sofern deren Granularität übereinstimmt. Dies ist für die verschiedenen Versionen derselben Familie zu erwarten.

Die Ausarbeitung der gemeinsamen Grundlage wird als Kommunikationsstandards-ontologie (Communication Standard Ontology – CSO) extrahiert und separat zur Verfügung gestellt.

Die Fachdomäne hingegen stellt, wie in Abbildung 15 bereits grob erläutert, eine eigene Ontologie dar, die als Mediator für alle Kommunikationsstandards fungiert, damit nicht jeder Kommunikationsstandard auf jeden anderen abgebildet werden muss. Außerdem ist intendiert, dass die Kommunikationsstandards die Fachdomäne unter bestimmten Aspekten beschreiben. Der Ansatz dieser Arbeit beruht auf der Entwicklung einer geeigneten Struktur, die feingranular und detailreich genug ist, um über alle benötigten Konzepte zu verfügen. Auf diese wird dann ein entsprechendes Mapping vorgenommen. Prinzipiell kann hier jede Ontologie eingesetzt werden, die eine derartige Vorgehensweise ermöglicht. Bei der Recherche zu dieser Arbeit wurden mehrere Kandidaten für eine solche Ontologie gefunden und im Folgenden kurz diskutiert. Die Entscheidung fiel zugunsten ACGT, die „Advancing Clinico Genomic Trials on Cancer“ [ACGT], weil sie schon einen Großteil der benötigten Konzepte enthält. Allerdings sind einige dieser Konzepte nicht granular genug bzw. fehlen ganz, so dass hier Erweiterungen innerhalb des Namensraums ACGT-FO vorgenommen werden müssen.

Nach der Entwicklung der Ontologien sowohl für die Kommunikationsstandards als auch für die Fachdomäne werden diese Ontologien in einem daran anschließenden Schritt über ein Regelwerk (Mapping) miteinander verknüpft.



Wie aus den vorhergehenden Ausführungen ersichtlich ist, kann nicht davon ausgegangen werden, dass die einzelnen Versionen der jeweiligen Familien zueinander voll kompatibel und daher direkt austauschbar sind. Daher muss umgekehrt das zu entwickelnde Regelwerk auch dazu genutzt werden können, um zwischen den einzelnen Versionen derselben Familie (in verschiedenen Ausprägungen) zu vermitteln.

Damit ergibt sich insgesamt folgende Darstellung:

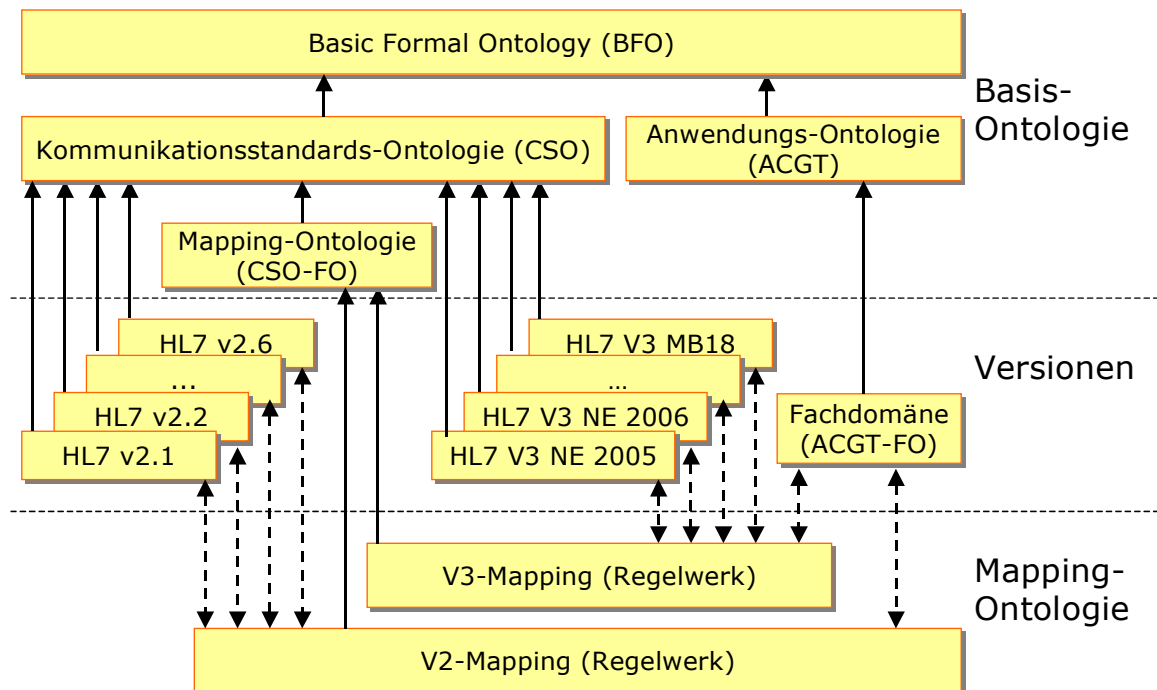


Abbildung 31: Ontologie-Struktur

Somit wird das Mapping jeweils familienintern auf einen gemeinsamen Standard – die Fachdomäne - reduziert. Letztere wird somit als gemeinsame Referenz genutzt, um ein übergreifendes Mapping zu ermöglichen.

### 10.2.2. Information Artefact Ontology

Zur Sicherstellung von Wissensaustausch und Interoperabilität sollten konsentrierte Ontologien soweit wie möglich wiederverwendet werden. Deshalb war es ein Gebot, vor Entwicklung einer eigenen Struktur nach bereits vorhandenen Ontologien zu recherchieren. Auf der Seite der OBO-Foundry [OBO] findet sich eine Reihe von Ontologien, von denen aber nur die IAO – die Information Artefact Ontology [IAO] – vom Namen und den mitgelieferten Beschreibungen her brauchbar erscheint. Die wichtigsten Klassen daraus sind nachfolgend in Protégé dargestellt:

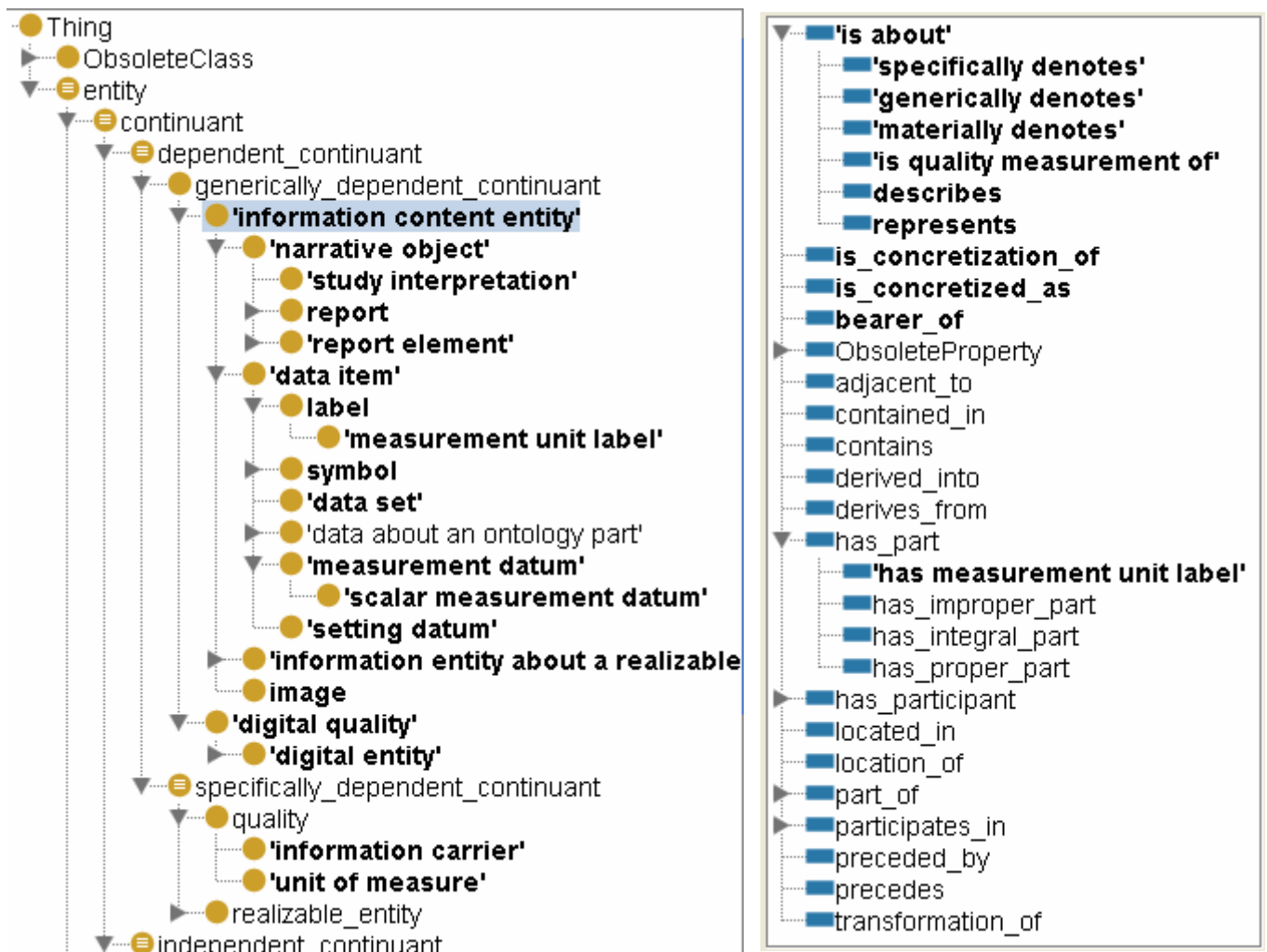


Abbildung 32: Information Artefact Ontology mit Relationen (Darstellung in Protégé)

Wie aus der Abbildung 32 hervorgeht, gibt es nur zwei Klassen, die für diese Arbeit in Frage kommen: "data item" und die Unterklasse "data set" [IAO]:

data item: *A data item is an information content entity that is intended to be a truthful statement about something (...) and is constructed/acquired by a method which reliably tends to produce (approximately) truthful statements.*

data set: *A data set is an information content entity that is an aggregate of other data items that have something in common and are considered to form a unit and are packaged together for some purpose, for example depositing into a repository.*

Mit diesen Definitionen können diese beiden Konzepte für Datenelemente sowie das Vokabular eingesetzt werden. Für alle anderen notwendigen Klassen gibt es keine Entsprechungen. Daher wird wegen der großen Diskrepanz auf den Einsatz dieser Ontologie generell verzichtet. Umgekehrt wird die Idee aber aufgegriffen, mit dieser Arbeit ebenfalls auf der Basic Formal Ontology [BFO] aufzusetzen.

### 10.2.3. Basis-Ontologie (Top-Level Ontology)

Wie ein einfacher Vergleich der Analyseergebnisse für die beiden HL7-Standards im vorhergehenden Abschnitt zeigt, basieren beide HL7-Familien auf denselben Grundlagen. Daraus kann in Phase 1 des Entwicklungsprozesses eine stark vereinfachte Basis-Ontologie generiert werden, wie der nachfolgende Screenshot von Protégé zeigt. In dieser werden die relevanten Konzepte mit abstrahierten Namen zusammengefasst, um sie später korrekt in BFO einzusortieren:

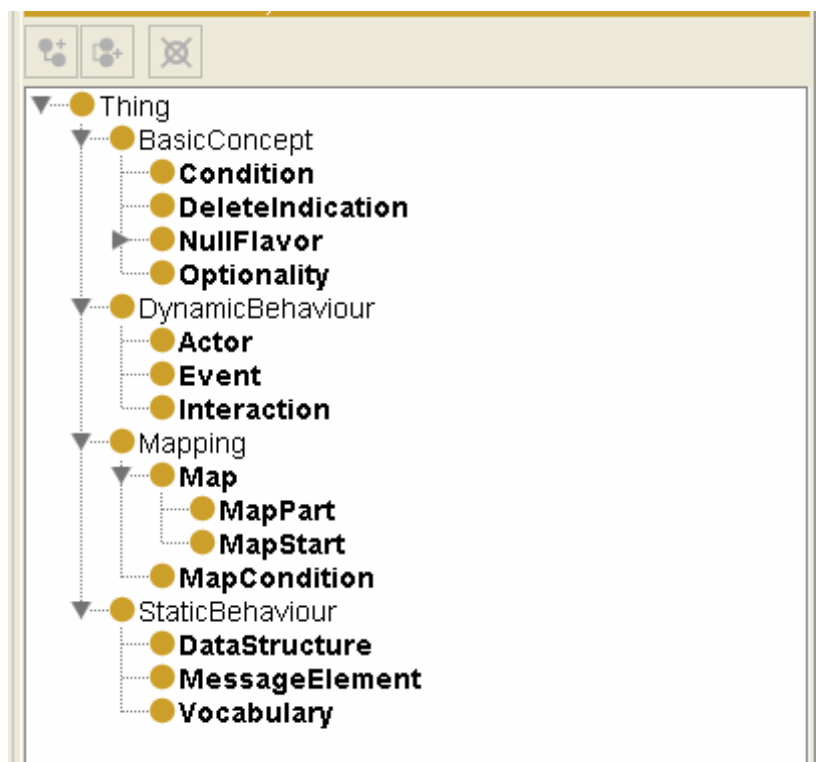


Abbildung 33: Basis-Ontologie: Konzepthierarchie (1. Entwurf als Darstellung in Protégé)

In einem zweiten Schritt sind die hier identifizierten Konzepte von der Referenz-Ontologie (BFO) abzuleiten. Damit werden Konzepte in den Einzelontologien für HL7 v2.x und V3 nicht direkt von owl:Thing abgeleitet, sondern sind jeweils Spezialisierungen von den hier aufgeführten Konzepten. Auch werden die hier verwendeten Klassennamen dann noch einmal überarbeitet.

Da das Werkzeug Protégé keine Doppeldefinitionen zur Lösung des *Cofounding Conflicts* (verschiedene Klassen mit demselben Namen/Label) zulässt, sind die Konzepte über entsprechende Namenskonventionen (Prä- und Postfix sowie Namespace) eindeutig zu machen. Diese können dann auch über verschiedene OWL-Dateien verteilt werden. Auf die daraus entstehende Problematik wird später noch detaillierter eingegangen.

Wie bereits angeführt, werden die hier identifizierten Konzepte später in die Klassenhierarchie von BFO eingeordnet. Deshalb sind die hier vorgestellten Benennungen und Strukturen auch nicht als final zu betrachten.

Auf die Relationenhierarchie wird an dieser Stelle noch nicht eingegangen.

#### 10.2.4. RO : Relationship Ontology

Ein Grundprinzip der OBO-Foundry ist die Wiederverwendung bereits vorhandener Ontologien. Dadurch entsteht eine Hierarchie von Ontologien, die über Import-Anweisungen eingebunden werden. Eine der zu verwendenden Basis-Ontologien ist dabei die Relationship-Ontology (RO), die keine eigenen Konzepte definiert, sondern ausschließlich Relationen (ObjectProperties). (Die bereits angeführte IAO definiert beispielsweise aufbauend auf RO zusätzliche eigene Relationen.)

Die nachfolgende Abbildung reflektiert alle definierten Relationen.

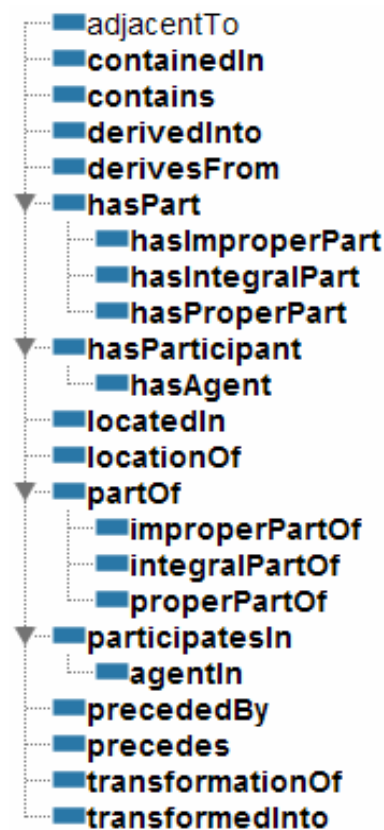


Abbildung 34: RO: ObjectProperties

Ein Teil der gelisteten Relationen stellt die Umkehrrelation (inverseRelationship) zu jeweils einer anderen Relation dar (beispielsweise „hasProperPart“ zu „properPartOf“). Insofern reduzieren sich die aufgelisteten Relationen sehr stark. Eine Nutzung der definierten Relationen ist aufgrund der fehlenden Dokumentation schwierig. Deshalb werden die für CSO notwendigen Relationen unabhängig von der RO definiert, eine spätere Angleichung wird aber nicht ausgeschlossen.

### 10.3. Diskussion konkurrierender Ansätze für eine Repräsentation von HL7-Kommunikationsstandards

Es gibt eine Reihe existierender, ggf. konkurrierender Ansätze für eine Repräsentation von HL7 Kommunikationsstandards in OWL, die sich allerdings alle auf eine Auswahl von



Autor	Ansatz
	<ul style="list-style-type: none"> <li>• ausgeliefert wird eine RDFS Datei, die aber nach OWL konvertiert werden kann</li> <li>• die Ontologie nutzt Object-Properties</li> <li>• Value-Sets sind als Spezialisierungen der Tabellen modelliert</li> <li>• inverse-Properties für Relationen sind vorhanden</li> <li>• eine bedingte Nutzung von Kardinalitäten ist vorhanden</li> <li>• rdfs:comment wird für die Beschreibung genutzt</li> <li>• es ist unklar, worüber "required" in OWL gesetzt wird: bei RDFS geschieht dies über minCardinality und maxCardinality. von daher geht bei der Konvertierung nach OWL etwas verloren, da „required“ eine besondere Bedeutung hat.</li> <li>• Properties sind z.T. inkonsistent, beispielsweise ActStatus.normal</li> </ul>
Tu	Samson Tu [Tu2008] hat 2 getrennte Ontologien für Datentypen und RIM entwickelt, die aber auf N3 [N3] – anstelle von OWL - basieren. Die Properties sind nicht sehr ausgeprägt und enthalten nur wenige Details und keine Vokabularien.
Bicer	Die generierten XML-Schemas (.xsd) werden nach OWL konvertiert. Auf diesen Dateien setzt dann ein spezifisches Mapping auf. Im Prinzip ist das nichts anderes als die Funktionsweise eines Kommunikationsservers – nur mit anderen Mitteln [Artemis, Bice2005, OemBlo2009a]. Somit muss ein Mapping für jede einzelne Nachrichten-Instanz wiederholt werden. Eine Wiederverwendung von Zwischenergebnissen ist nicht möglich.

Lediglich Bicer gibt an, für welchen Zweck die Ontologien entwickelt und realisiert wurden. Das ist aber wesentlich für das Design der Ontologien. Diese Dissertation verfolgt einen völlig anderen und zugleich umfassenderen Ansatz.

#### **10.4. Die Communication Standards Ontology als Referenz-Ontologie**

Die vorhergehenden Überlegungen werden nachfolgend genutzt, um eine gemeinsame Referenz-Ontologie abzuleiten. Dazu wird das GCM eingesetzt, um die gemeinsamen Strukturen aufzuzeigen und daraus sowohl die notwendigen Konzepte als auch die dazugehörigen Relationen herzuleiten und somit Konsistenz zu gewährleisten. Die Details werden nachfolgend erläutert.

##### **10.4.1. Top-Level-Ontologie mit nächster Ebene**

Werden in der Abbildung 33 die nächsten Ebenen mit in die Darstellung einbezogen, so ergibt sich folgendes Bild:

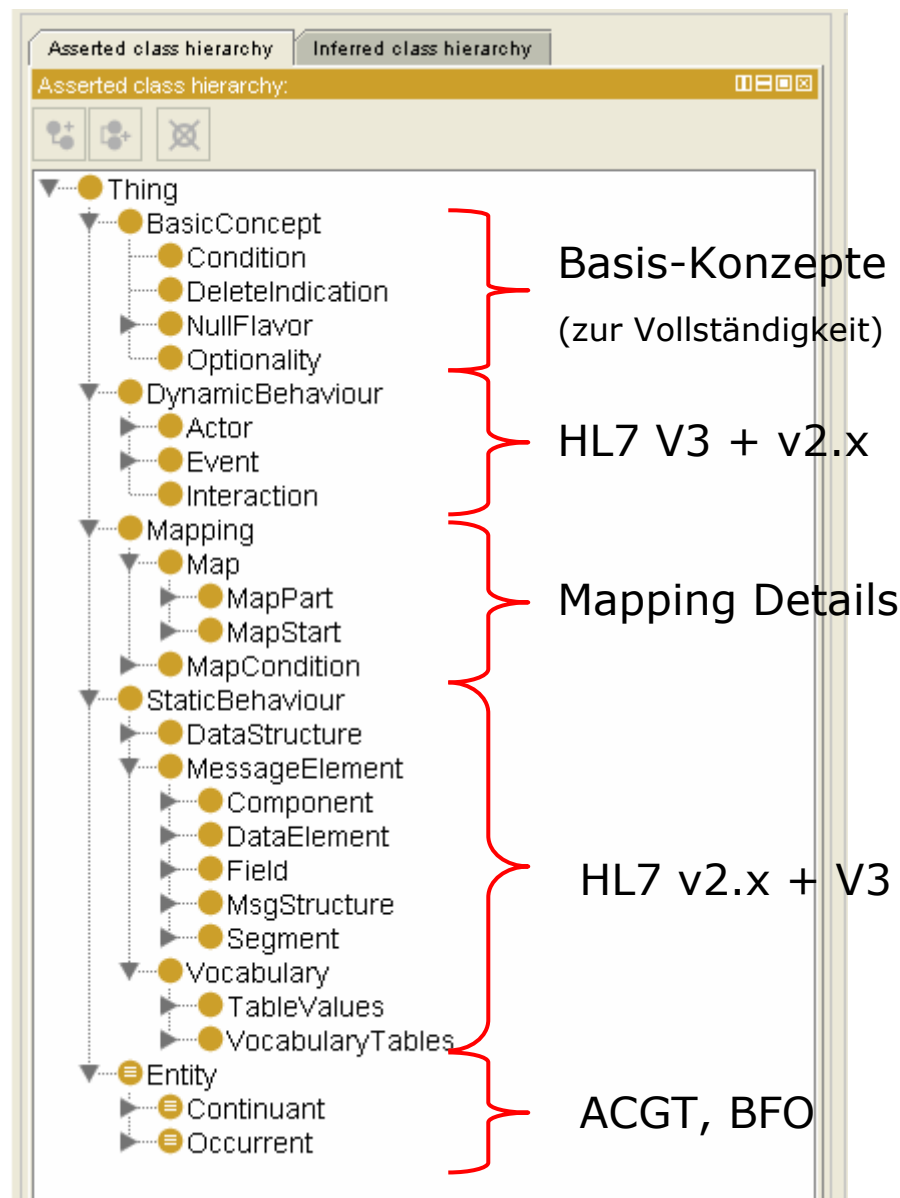


Abbildung 36: Erweiterte Top-Level Ontologie (Darstellung in Protégé)

In dieser Darstellung sind auch die obersten zwei Ebenen von BFO (und ACGT) enthalten: Entity mit Continuant und Occurrent. (Die Einsortierung der Klassen in die BFO-Hierarchie ist in dieser Abbildung jedoch noch nicht erfolgt.) Ebenso sind die ersten Überlegungen zum Mapping zu separieren. Beides wird in nachgelagerten Kapiteln detaillierter erläutert und weiter verfeinert.

### 10.4.2. CSO + BFO: Hierarchie

Im nächsten Schritt gilt es, diese Klassen in BFO einzusortieren:

Tabelle 7: CSO + BFO

Element (Klasse)	Erläuterung
Thing	
entity (bfo)	

Entwicklung einer ontologiebasierten Architektur zur Sicherung semantischer Interoperabilität zwischen Kommunikationsstandards im Gesundheitswesen

Element (Klasse)	Erläuterung
continuant (bfo) dependent_continuant (bfo) generically_dependent_continuant (bfo)	
InformationObject (cso)	Für dieses Konzept könnte das Konzept aus ACGT genutzt werden
DataStructure (cso) Document (cso)	
MessageElement (cso) Message (cso) Vocabulary (cso)	
specifically_dependent_continuant (bfo) quality (bfo)	
InformationObjectQuality (cso)	
realizable_entity (bfo) disposition (bfo) function (bfo) role (bf)	
Actor (cso)	
independent_continuant (bfo)	
occurent (bfo) processual_entity (bfo) fiat_process_part (bfo) process (bfo)	
DocumentCreation (cso) Interaction (cso) MessageExchange (cso)	
process_aggregate (bfo) process_boundary (bfo)	
Event (cso)	
processual_context (bfo) spatiotemporal_region (bfo) temporal_region (bfo)	

In Protégé stellt sich das dann wie folgt dar, wobei die neu eingeführten Konzepte<sup>25</sup> fett markiert sind:

<sup>25</sup> Die Namensgebung stößt auf Probleme, da die Namen automatisiert aus den vorliegenden Daten geniert werden, so dass die Namen/Label mitunter unpassend erscheinen. Das einfachste Mittel zur Umgehung sind geeignete Prä- oder Postfixe, die dann an die Label angefügt werden. Dieses Vorgehen wäre sicherlich die bessere Wahl, allerdings erschwert dies die Lesbarkeit der Abbildungen und Tabellen, so dass darauf verzichtet wird.

Ein Beispiel wäre das „LivingSubject“, das im Falle von HL7 V3 nur eine Spezialisierung des InformationObjects ist.



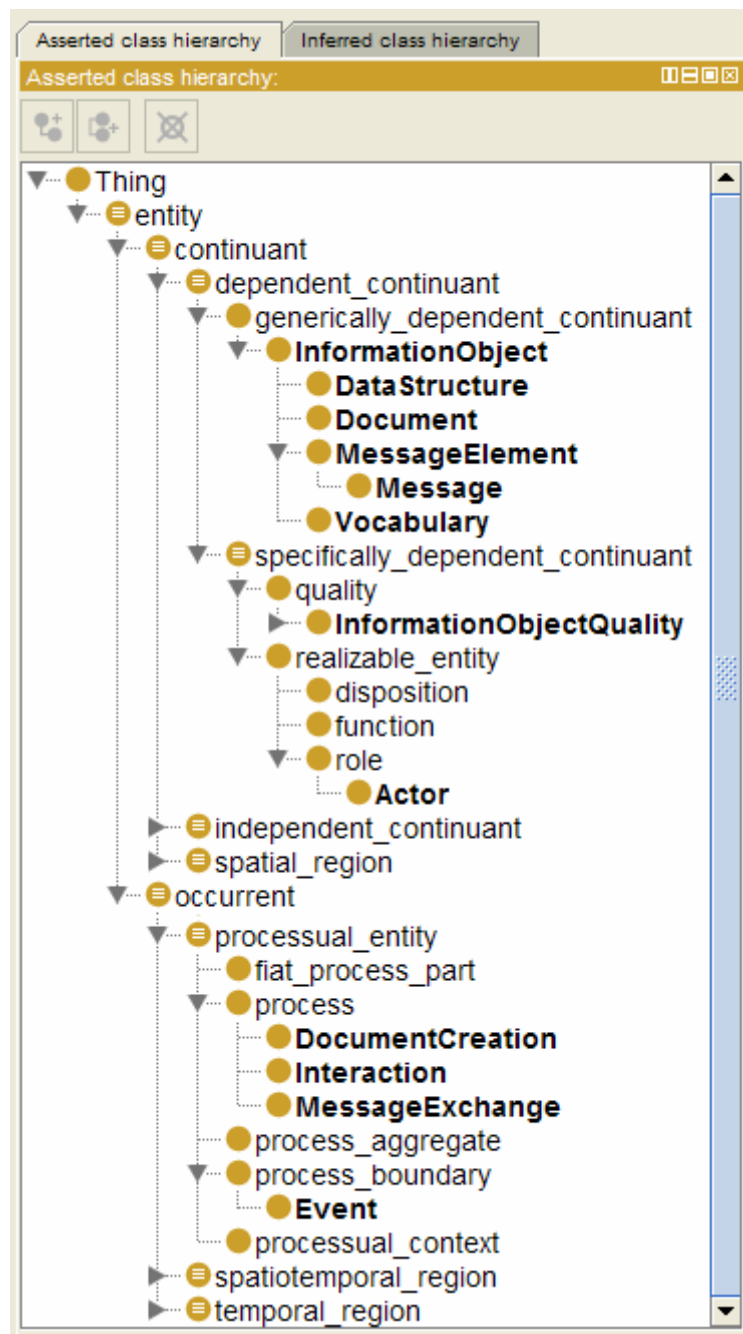


Abbildung 37: CSO (Darstellung in Protégé)

### 10.4.3. Eigenschaften der Informationsobjekte (als Qualitäten)

Datenelemente, Felder, Komponenten etc. haben bestimmte Eigenschaften, die sich am besten als eigenständige Konzepte modellieren und dann in OWL umsetzen lassen.

Diese Eigenschaften sind in der nachfolgenden Abbildung aufgeführt.

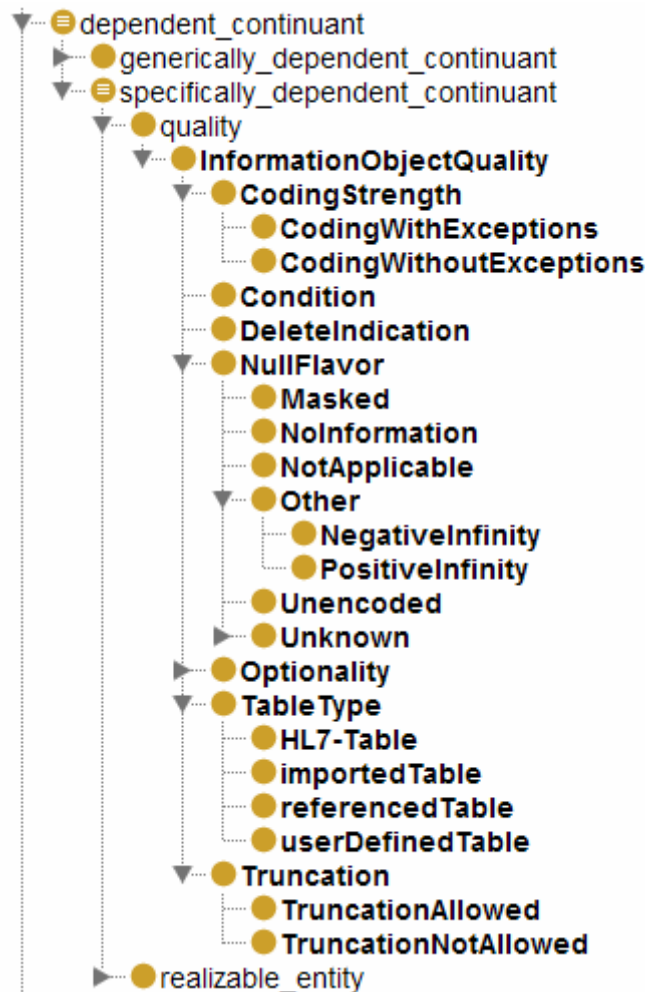


Abbildung 38: Basis-Konzepte von CSO (Darstellung in Protégé)

Über entsprechende Relationen (ObjectProperty) können die Spezialisierungen des InformationObject direkt mit diesen Qualitäten verbunden werden.

Da diese Qualitäten jeweils als eigene Klasse repräsentiert sind und nicht als Data-Property, lassen sie sich viel einfacher auswerten. Dieser Aspekt wird in der abschließenden Diskussion wichtig, wenn ein Interpretier/Agent diese Wissensbasis nutzen soll.

Die Bezeichnungen (Label) in vorhergehender Abbildung richten sich nach den Namen in den Standards. Aus OWL-Sicht sind diese Namen damit nicht unbedingt optimal, da sie Substrings wie „Type“ enthalten. (Dasselbe Problem taucht später bei den Spezialisierungen der Informationsobjekte wieder auf.) Es kann hier aber nicht die Aufgabe dieser Arbeit sein, dieses Problem ursächlich zu lösen. Daher werden diese Bezeichnungen übernommen.

#### 10.4.4. Optionalität

Eine der aufgelisteten Eigenschaften ist die "Optionalität". Wie im IHE Whitepaper [IHE Vol.0] erläutert ist, haben die verschiedenen Kommunikationsstandards hier unterschiedliche Interpretationen desselben Konzepts, das in den Spezifikationen über einen

Term (beispielsweise "Required") ausgedrückt wird. In der nachfolgenden Übersicht sind die Terme aus

- HL7 v2.x,
- HL7 V3 und
- DICOM<sup>26</sup>

enthalten und werden in OWL durch eindeutige Namespaces (beispielsweise "hl7-v2" oder "hl7-v3") unterschieden. Eine Gleichheit der Konzepte wird in der nachfolgenden Abbildung über das spezielle Gleichheitszeichen visualisiert. (Leider ist der zugeordnete Namespace nicht erkennbar<sup>27</sup>.)

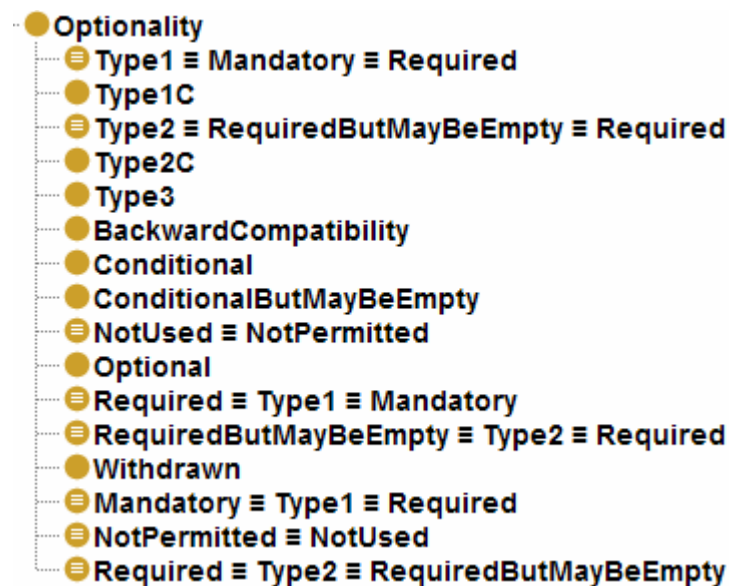


Abbildung 39: Optionalität (Darstellung in Protégé)

In dieser Aufstellung fällt beispielsweise auf, dass das Konzept "Required" (in obiger Grafik als Klasse dargestellt) zweimal enthalten ist, aber unterschiedliche Äquivalenzrelationen hat. Dies lässt sich damit erklären, dass es sich genaugenommen um zwei verschiedene Konzepte handelt, obwohl sie in den beiden Standards gleich benannt sind. Umgekehrt gibt es in den Standards unterschiedlich benannte Konzepte, die aber dasselbe meinen/bedeuten: "DICOM Type 1" ≡ "HL7 V3 Mandatory" ≡ "HL7 v2 Required".

<sup>26</sup> DICOM (Digital Imaging and COmmunication in Medicine [DICOM]) wird hier in die Betrachtung mit einbezogen, weil dieser Standard im Bereich der Optionalitäten wieder eigene Begriffe einführt, die durchaus mit den bereits vorhandenen zur Deckung gebracht werden können. Von daher kann DICOM an dieser Stelle als *Proof-of-Concept* für die Allgemeingültigkeit des Ansatzes verstanden werden, ohne jetzt näher ins Detail gehen zu wollen. Dafür sei auf die entsprechende Literatur verwiesen.

<sup>27</sup> Dies kann man in Protégé einstellen. Allerdings wird die Anzeige nicht wie gewünscht aufgebaut. Es scheint deshalb ein Fehler in Protégé zu sein, da es mit 3.4 geht. Es soll hier aber die Einheitlichkeit in der Darstellung gewahrt bleiben, so dass auf dieses Detail verzichtet wird.

### 10.4.5. Beziehungen/Relationen (Object-Properties)

Beziehungen können in OWL mittels Object-Properties ausgedrückt werden. Nachfolgend werden die Relationen von CSO dargestellt, wobei die Relationen von BFO ausgeblendet sind. Diese Relationen bilden derzeit eine eigene Hierarchie, d.h. die Relationship Ontology (RO) wird nicht genutzt. Dies müsste in einem weiteren Schritt nachgeholt werden.

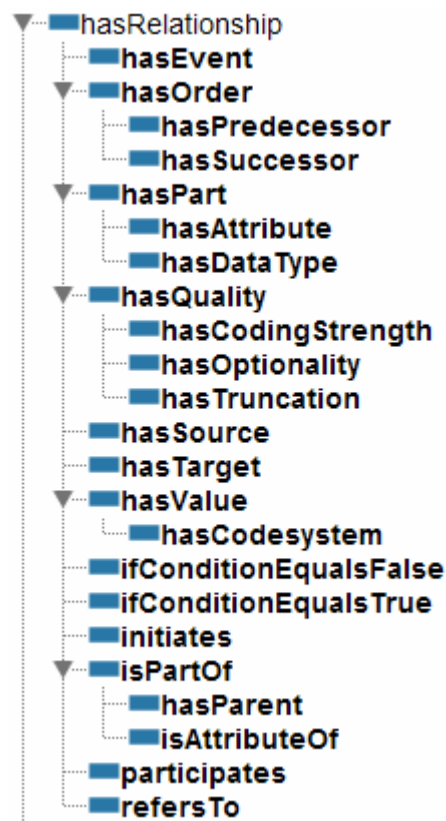


Abbildung 40: Relationen (Darstellung in Protégé)

Die zuvor aufgelisteten ObjectProperties bedürfen teilweise noch einer detaillierteren Definition:

hasQuality	fasst die verschiedenen ObjectProperties zusammen, um den InformationObjects spezielle Qualitäten zuzuweisen.
hasSource	inverse-of hasTarget wird genutzt, um zwei InformationObjects direkt in Beziehung zu setzen.
hasDataType	Die Datentypen sind komplexe Konzepte und enthalten zusätzliche Informationen, die als selbständige Konzepte dargestellt werden sollen. (Von der Semantik entspricht dies einer Part-of-Relation.) Das OWL DatatypeProperty eignet sich deshalb hier nicht. Stattdessen bietet sich eine Umsetzung als eigene Klasse an, die dann über dieses ObjectProperty mit dem korrespondierenden InformationObject in Beziehung gesetzt

	wird.
hasAttribute	Wie hasDataType, nur auf Attributebene.
isPartOf	inverse-of hasPart wird genutzt, um den Aufbau von Komponentenstrukturen über OWL zu ermöglichen.
ifConditionEqualsTrue	werden im Mapping-Prozess benötigt, um Bedingungen aus-
ifConditionEqualsFalse	werten zu können. Über diese beiden ObjectProperties wird dann auf die Mapping-Konzepte/Klassen verwiesen, mit denen bei (Nicht-)Zutreffen der Bedingung weitergemacht werden soll.
hasOrder	Hierarchische Nachrichten müssen serialisiert werden, um eine Übertragung zu ermöglichen. Die hierfür notwendigen Strukturinformationen werden über diese ObjectProperties hinterlegt.
hasValue	Wird genutzt, um einen speziellen InformationObject einen konkreten Wert zuzuweisen. Diese werden ebenfalls in Form von Konzepten ausgedrückt.

Relationen zwischen zwei Konzepten werden über ObjectProperties ausgedrückt, daher sollte auch festgelegt werden, von welchem Teilbaum der Konzepthierarchie zu welchem anderen eine solche Beziehung konkret aufgebaut werden kann. Dies geschieht durch die Angaben „domain“ (von) und „range“ (nach).

#### 10.4.6. Data-Properties

In CSO werden keine DatatypeProperties genutzt, da der Ausmodellierung über separate Konzepte der Vorrang gegeben wird.

#### 10.4.7. Instanzen

Es kommen keine Instanzen in den Ontologien vor, da für die Modellierung nur Mengen (T-Box) benötigt werden.

### 10.5. Abbildung HL7 V3 $\Rightarrow$ OWL

#### 10.5.1. Datenquelle

Für die Umsetzung von HL7 V3 in eine OWL-basierte Ontologie bieten sich aufgrund der Rohdaten (s. nachf. Abb.) verschiedene Verfahren an:

Tabelle 8: Umsetzungsmöglichkeiten für V3 je nach Datenquelle

Quelle	Datenformat	Zugriff	Kommentar
Pubs-DB	relationale Datenbank	ODBC oder XML-Export	wird in Zukunft durch MIF abgelöst
MIF	XML	DOM mit VBA oder XSLT	Wird in verschiedenen Versionen bereitgestellt. In dieser Arbeit

Quelle	Datenformat	Zugriff	Kommentar
			wird die Version 2.0 verwendet.
XML	XML	DOM mit VBA oder XSLT	nur temporäres Zwischenformat

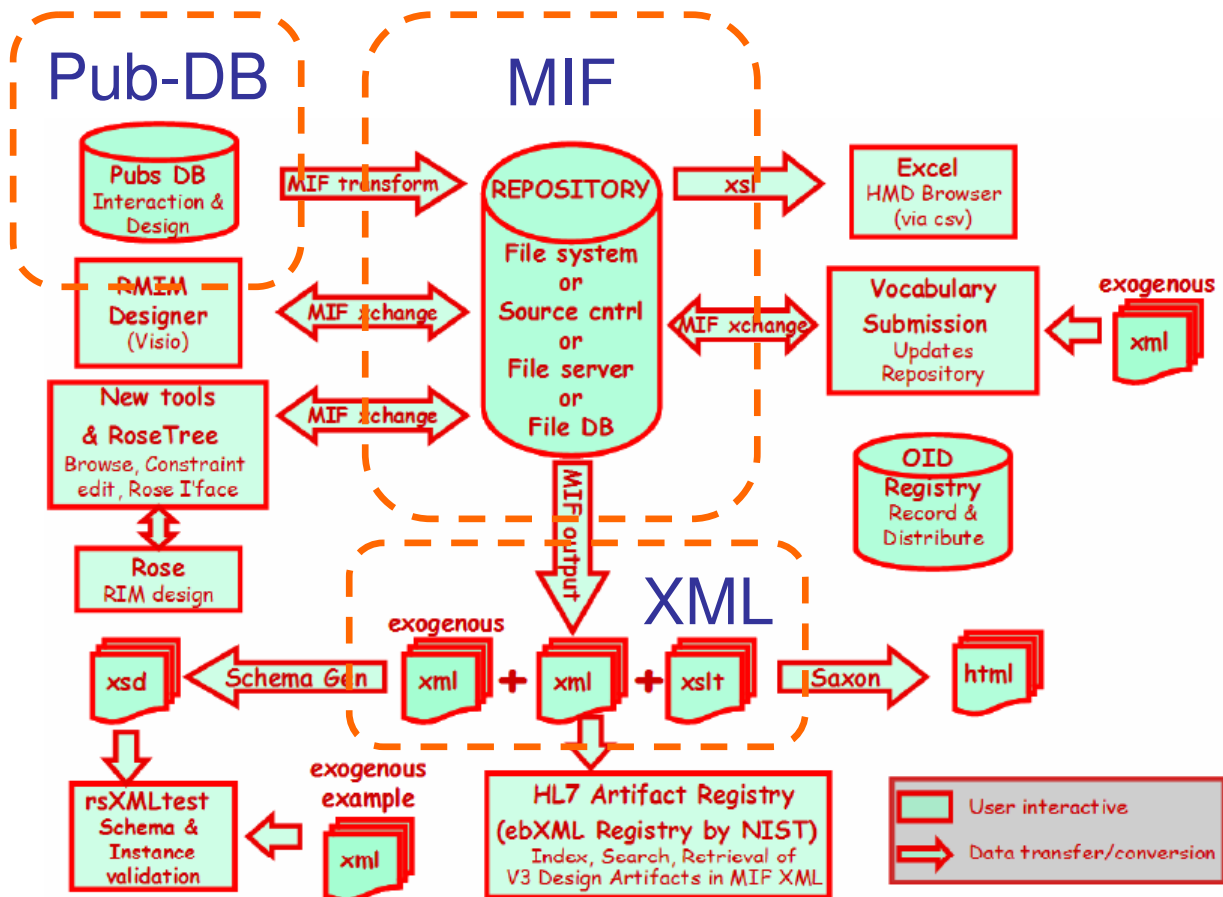


Abbildung 41: HL7-V3-Tooling-Sicht (nach McKenzie)

In Anbetracht der Tatsache, dass die HL7-Community beabsichtigt, in Zukunft auf das sog. MIF-Format zu setzen, werden die diversen Abbildungen nach OWL auf dieser Grundlage evaluiert. Da das eigene MIF-Format auf XML basiert, sollten XSL-Transaktionen das Mittel der Wahl sein. Die Arbeiten haben aber gezeigt, dass hierdurch recht komplexe XSLTs entstehen, die nicht ohne weiteres wartbar sind. Daher bietet sich alternativ ein kleines Programm auf Java- oder VB-Basis an. Durch die hiermit verfügbare Funktionalität (beispielsweise direkte Sichtenbildung mit Sortierung) wird der notwendige Entwicklungsaufwand drastisch reduziert. (Eine Evaluierung der Möglichkeiten zur Generierung der Ontologien findet sich in Kap. 14.2.1. Scripting)

### 10.5.2. Einordnung in das GCM

Grundlegendes Design-Prinzip des GCM ist die Vereinfachung von Systemen durch Dekomposition. Die Einschränkung auf eine einzelne Domäne – in diesem Fall ist es HL7 V3 – mit einer feingliedrigeren Unterteilung im Domänenbereich und der gleichzeitigen Einschränkung auf Enterprise und Information Views führt zu nachfolgender Darstellung:

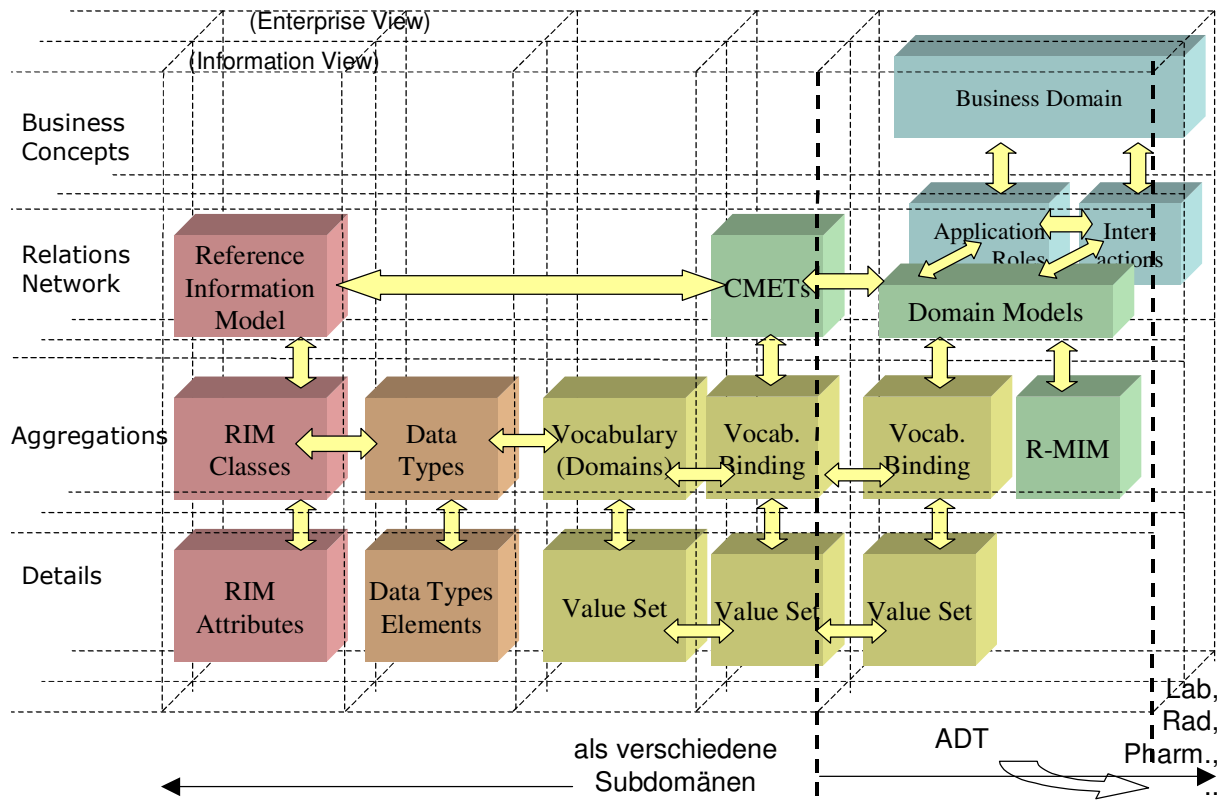


Abbildung 42: Darstellung von HL7 V3 im GCM

Aus dieser Darstellung lassen sich folgende Schlussfolgerungen bzw. Bemerkungen ableiten:

- Für ein Mapping auf HL7 v2.x werden "Events" benötigt, die allerdings nur in Form der Interactions definiert sind. In HL7 V3 werden diese zusammen mit den Application Roles angegeben. Beide sind aber nur informal definiert und gehören in den Enterprise View.
- Die den Application Roles und Interactions übergeordnete Aggregation ist in HL7 V3 nicht definiert. Für eine korrekte und vollständige Modellierung wäre dies aber notwendig.
- Die CMETs stellen Modelle dar, die domänenübergreifend gelten und in einem "Realm" (Einsatzgebiet) gemäß den lokalen Anforderungen abgewandelt werden können.
- Es bietet sich an, die Informationen auf der Detail-Ebene ebenfalls als eigene Klassen umzusetzen, da hier detailreiche Beziehungen zu anderen Informationen auf derselben Ebene bestehen.
- Die CMETs und die Domänen-Modelle (D-MIMs) schränken die Nutzung der Vokabularien ein. Aus diesem Grund enthält obige Darstellung ein Vocabulary Binding, das seinerseits sowohl in Beziehung zur Vokabeldomäne als auch zum Value Set steht.
- Zur Vereinfachung sind nicht alle Beziehungen eingezeichnet. Insbesondere in der

horizontalen Dimension gibt es weitere Bindungen. So müsste beispielsweise das D-MIM ebenfalls eine Beziehung zum RIM haben.

- g) Die rechte Spalte – hier stellvertretend als ADT dargestellt – müsste für alle Anwendungsdomänen (beispielsweise Labor, Radiologie, Scheduling, Pharmacy) wiederholt werden. Dies ist durch die Auflistung in der Graphik unten rechts angedeutet.
- h) Aus dieser Darstellung lässt sich die Import-Struktur für die Ontologien ableiten.

### 10.5.3. Darstellung der Informationen

Die OWL-Spezifikation für HL7 V3 muss für alle RIM-Klassen realisiert werden, um eine ausreichend fundierte Grundlage zu haben. Die in obiger Form angegebenen XML-Headerinformationen werden in der nachfolgenden Auflistung der Einfachheit halber weggelassen. Die "*structural attributes*" werden dabei direkt über Eigenschaften/Relationen (als ObjectProperties) dargestellt:

Für die anderen Klassen sowie die weiteren Spezialisierungen geschieht dies entsprechend. Auf diese Weise werden die in den verschiedenen HL7-Modellen<sup>28</sup> (D-MIM, R-MIM, HMD<sup>29</sup>, CMETs) enthaltenen Elemente über ihren Typ und damit in verschiedenen Teilbäumen repräsentiert. Zur bidirektionalen Abbildung in die bisher spezifizierten ITS (Implementation Technology Specification) benötigen die Elemente den dort verwendeten Namen. Insgesamt wird eine relativ flache Struktur aufgebaut, die einerseits die Verwaltung vereinfacht und andererseits als Einsprungpunkt für Constraints dienen kann.

### 10.5.4. OWL-Elemente für HL7 V3

Die nachfolgend aufgelisteten Klassen sind über die verschiedenen Skripte erzeugt worden. Hierbei gilt es, verschiedene Problemfelder in den Griff zu kriegen:

- Die einzelnen Klassen müssen jeweils zueinander disjunkt sein, allerdings wird dies in der Generierung noch nicht berücksichtigt. Es sollte überlegt werden, inwieweit hier die Ontology Preprocessor Language (OPPL) [OPPL] nachträglich helfen kann.
- Die Spezialisierungshierarchie erscheint nicht immer selbsterklärend bzw. logisch/intuitiv.
- Namenskonventionen sind erforderlich, um den Generierungsprozeß bzw. den Automatismus zu unterstützen und Konflikte zu vermeiden.

Die letzten beiden Punkte werden insbesondere bei den Datentypen ersichtlich.

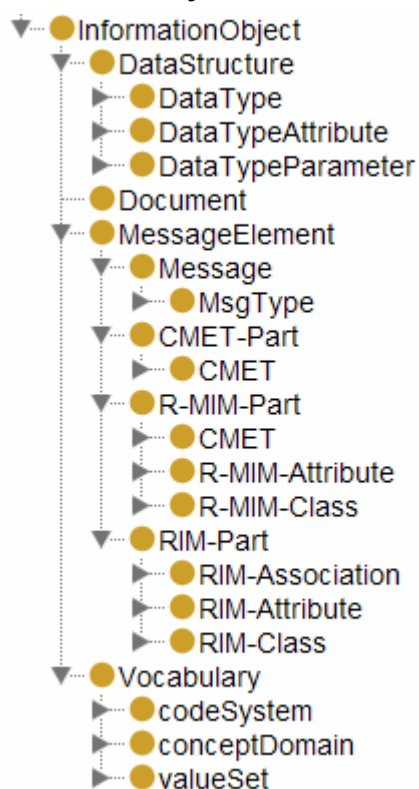
---

<sup>28</sup> An dieser Stelle sei auf die Homepage verwiesen, auf der eine einführende Erläuterung zu finden ist (s. Kap. 1. Bereitstellung der Ergebnisse).

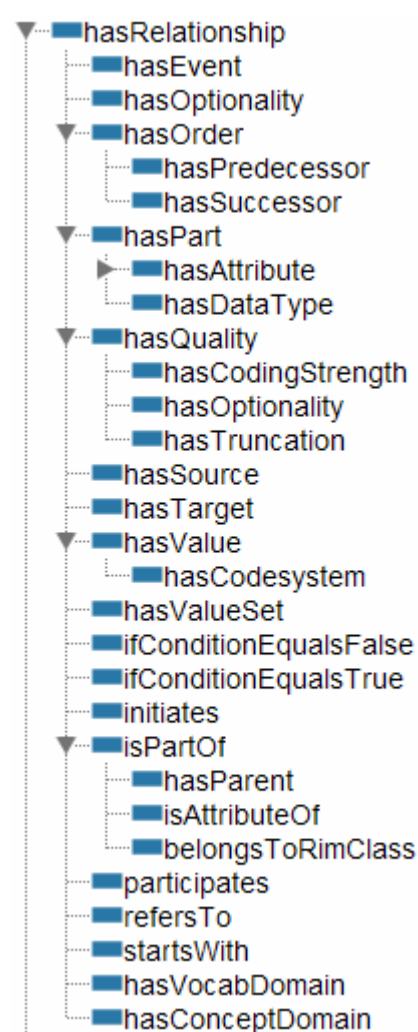
<sup>29</sup> HMDs sind von der Struktur her ebenfalls R-MIMs, jedoch mit zusätzlichen Einschränkungen, so dass sie in der weiteren Betrachtung nicht berücksichtigt werden müssen.



## InformationObject

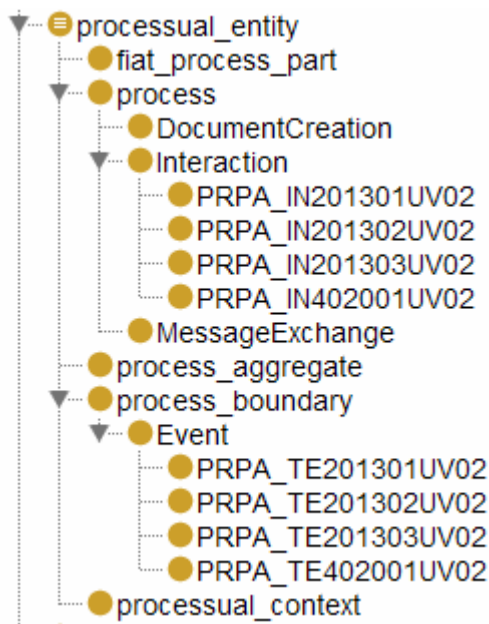


## Relations

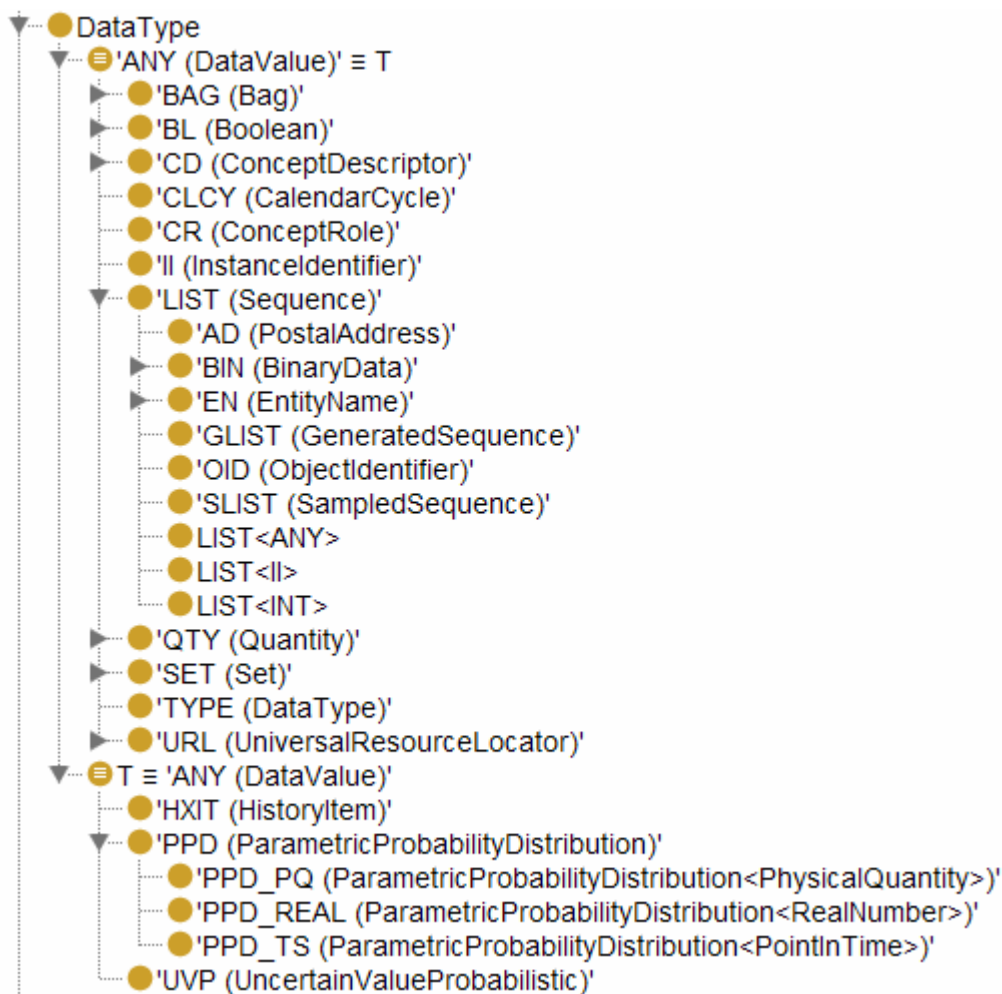


Nachfolgend werden die einzelnen Konzepte im Detail weiter ausgeführt. Hierbei ist nochmals anzumerken, dass die Konzeptnamen (Label) durch den Generierungsprozess und die MIF-Dateien vorgegeben sind. Aus Gründen der Handhabung sind teilweise Zwischenhierarchien eingezogen worden, die nicht unbedingt notwendig sind, aber das Arbeiten mit Protégé deutlich erleichtern. Dazu gehört auch die Verifikation der Ergebnisse. Derartige Zwischenstufen in den Hierarchien sind ebenfalls als zusätzliche Referenzen für Konsistenzbedingungen (Reasoning) hilfreich. In den nachfolgenden Abbildungen sind einige Zwischenstufen ausgeklappt“, um die interessanten Details sichtbar zu machen.

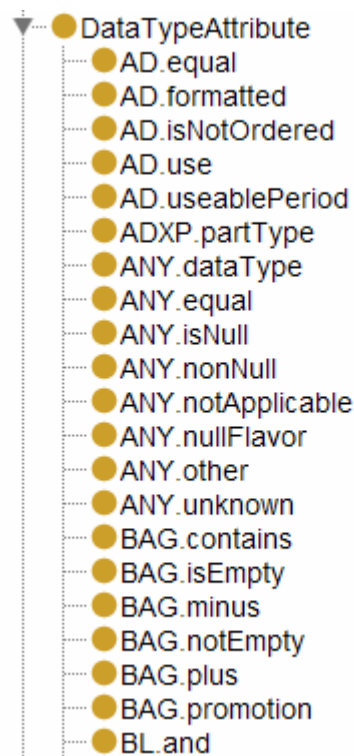
Interaction + Event



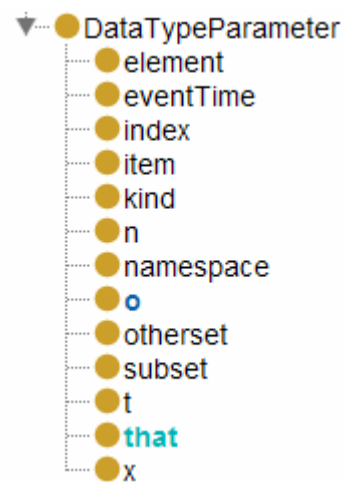
DataType



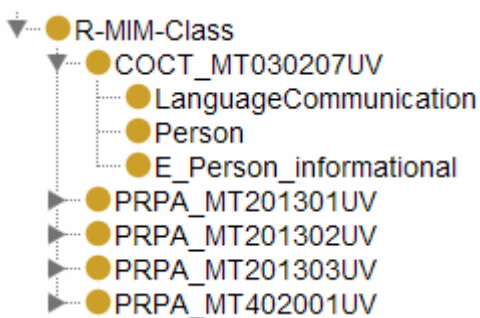
### DataTypeAttribute<sup>30</sup>



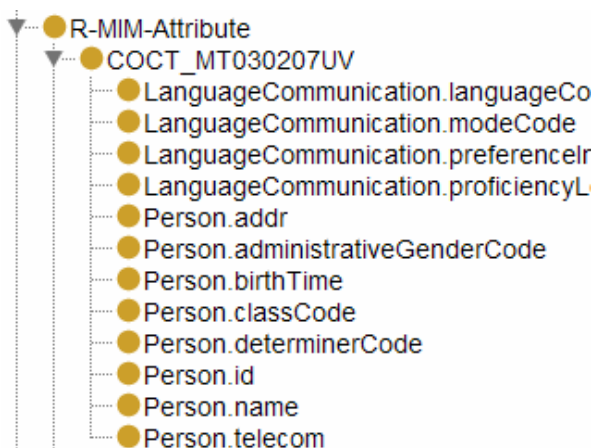
### DataTypeParameter<sup>31</sup>



### R-MIM-Class



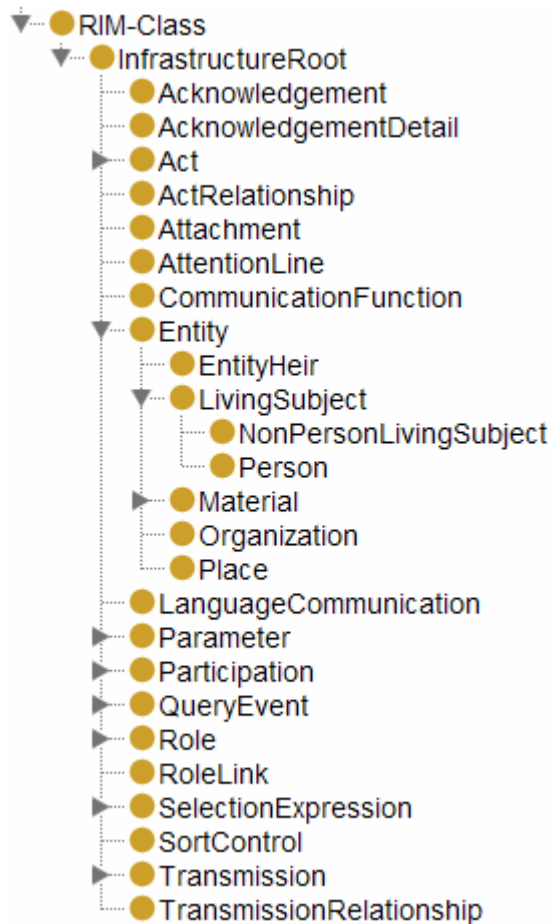
### R-MIM-Attribute



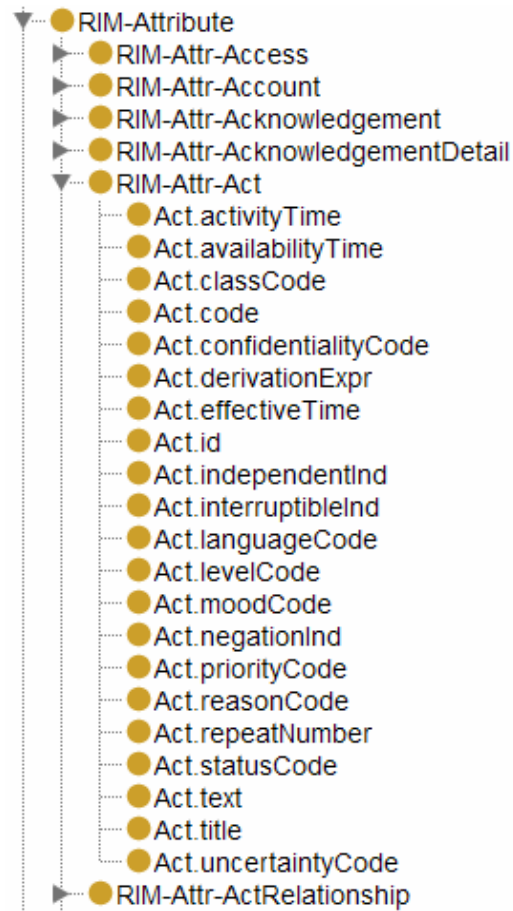
<sup>30</sup> Hier wäre beispielsweise zu erwägen, eine weitere Zwischenstufe (beispielsweise „AD-DataTypeAttribute“ und „BAG-DataTypeAttribute“) einzuführen, um das Arbeiten mit Protégé zu erleichtern.

<sup>31</sup> Einzelne Klassen sind durch Protégé in dieser Grafik farblich markiert worden, da es sich um besondere Schlüsselworte handelt.

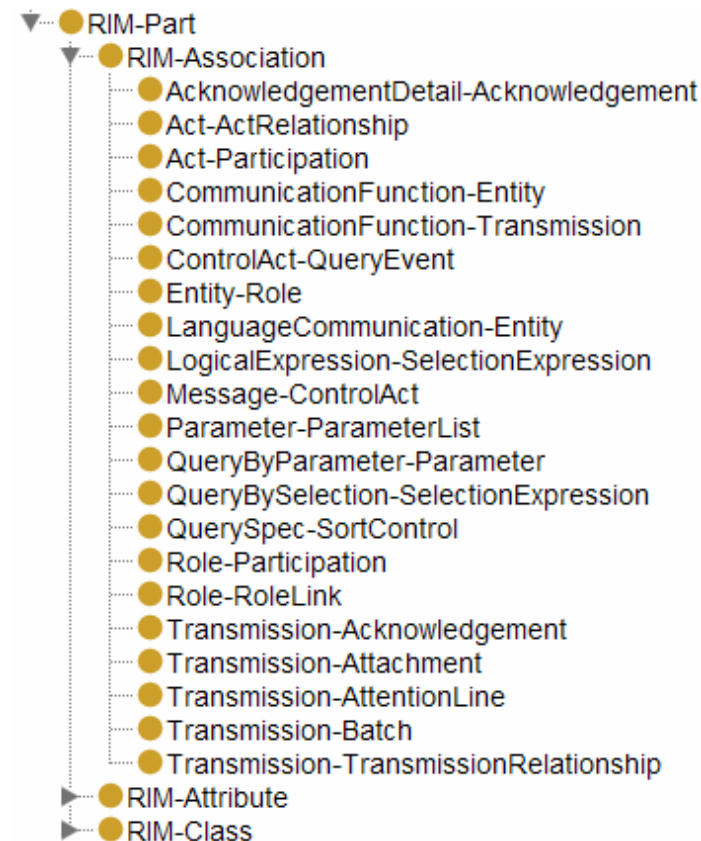
RIM-Class

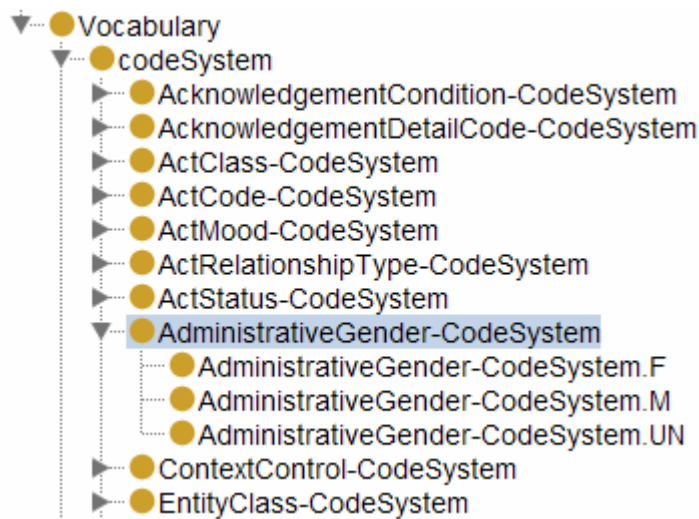


RIM-Attribute

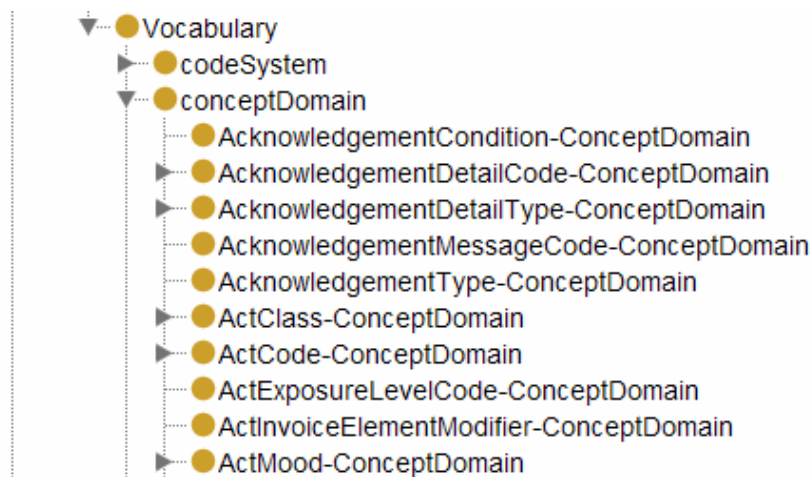


RIM-Association



Vocabulary codeSystem<sup>32</sup>:

## Vocabulary conceptDomain:



## Vocabulary valueSet

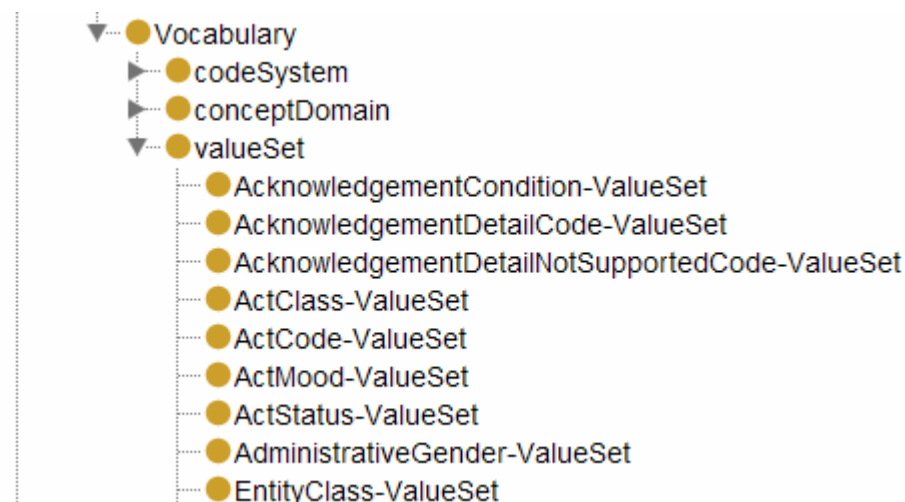


Abbildung 43: HL7-V3-Elemente in OWL (dargestellt in Protégé)

<sup>32</sup> Vorschläge für Namenskonventionen findet man beispielsweise in [SchKus2007].

Anm.: Wie bereits vorangehend erläutert, werden die Namen aus den MIF-Dateien generiert. Daher besteht kein direkter Einfluß auf die Namensgebung. [SchKus2007]

### 10.5.5. Fehler in den Quelldaten

In den MIF-Dateien für das RIM sowie den ausgewählten R-MIMs werden Vokabeldomänen genutzt (referenziert), die in der MIF-Datei für die Vocabularies nicht vorkommen. Auch wenn diese Fehler aufgrund falscher Referenzen (bedingt durch unterschiedliche und nicht aufeinander abgestimmte Dateien) entstehen und dann durch manuelle Prozesse teilweise korrigiert werden können, so handelt es sich doch um Inkonsistenzen, die in der Anzeige als direkte Spezialisierungen<sup>33</sup> von "owl:Thing" auffallen. Die nachfolgende Tabelle listet die für HL7 V3 MB#19 entdeckten Inkonsistenzen einschließlich der Korrekturen auf:

Tabelle 9: Fehler in der MIF-Definition des RIM 0219

Verwendet im RIM	in Vokabeldomäne vorhanden	Lösung
TransmissionRelationshipType	TransmissionRelationshipTypeCode	Ersetzen
SubstanceAdministrationMethod	ActSubstanceAdministrationCode	Ersetzen
ActContextLevel		nicht verwenden
AdministrableDrugForm		nicht verwenden

Tabelle 10: Fehler in der MIF-Definition des R-MIM

Verwendet im R-MIM	in Vokabeldomäne vorhanden	Lösung
x_ActEncounterReason	X_ActEncounterReason	Ersetzen
x_ActMoodIntentEvent	X_ActMoodIntentEvent	Ersetzen
x_EncounterAdmissionUrgency	X_EncounterAdmissionUrgency	Ersetzen
ActEncounterAccommodationCode	_ActEncounterAccommodationCode	Ersetzen
ServiceDeliveryLocationRoleType		nicht verwenden

Diese Beispiele demonstrieren den zusätzlichen Nutzen des gewählten Ansatzes zur Auffindung von Inkonsistenzen in den existierenden HL7-Spezifikationen.

## 10.6. Abbildung HL7 v2.x ⇒ OWL

### 10.6.1. Datenquelle

Die Erzeugung einer Ontologie für HL7 v2.x basiert auf der vom Autor generierten HL7-Datenbank, in der alle benötigten Informationen enthalten sind. Ein Programm auf Basis

---

<sup>33</sup> An dieser Stelle ist zu erwähnen, dass bei einer inkorrekten Auswahl des "ontologischen Subsets" derselbe Effekt auftritt, der dann aber durch eine Erweiterung der Auswahl im Generierungsprozess behoben werden muss.

von VBA extrahiert die notwendigen Daten und konvertiert diese nach OWL, so dass Protégé die Dateien einlesen und visualisieren kann:

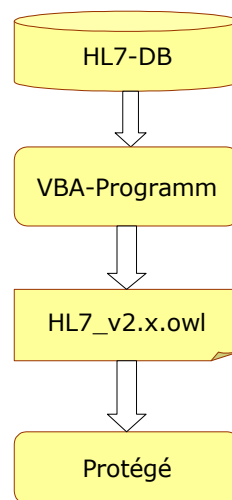


Abbildung 44: Ableitungsprozess der geforderten OWL-Spezifikationen für HL7 v2.x

### 10.6.2. Einordnung in das GCM

Wie zuvor HL7 V3, lässt sich HL7 v2.x ebenfalls in das GCM einordnen, allerdings ist es nicht so detailreich wie V3:

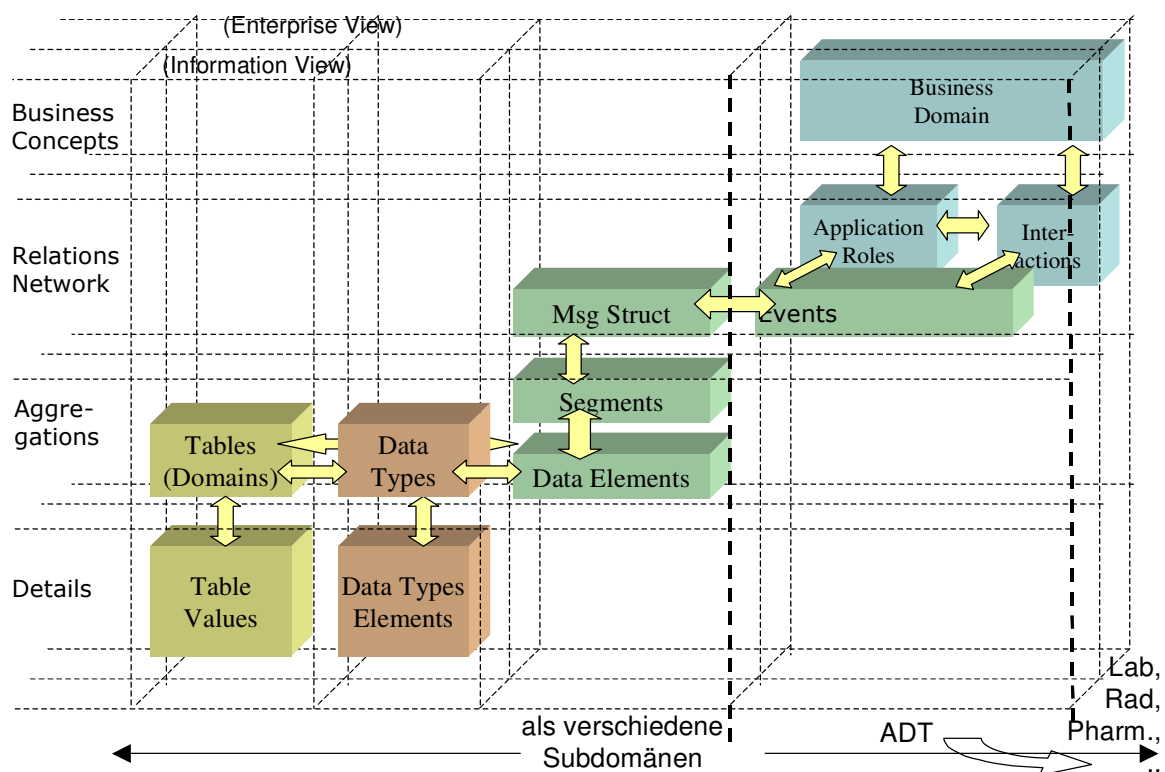


Abbildung 45: Einsortierung von HL7 V2.x in das GCM

Analog zu V3 lassen sich hier ebenfalls Schlussfolgerungen bzw. Bemerkungen ableiten:

a) Für ein Mapping auf HL7 v2.x werden "Events" benötigt. Die damit verbundenen

- Application Roles und Interactions sind allerdings nicht definiert. Beide gehören in den Enterprise View.
- b) Das den Application Roles und Interactions übergeordnete Level mit der Business Domain ist in HL7 v2 nicht definiert. Für eine korrekte und vollständige Modellierung wäre es aber ebenfalls notwendig.
  - c) Die Ereignisse sind direkt mit den Nachrichtenstrukturen verbunden, die zum Aufbau der konkreten Nachrichteninstanzen verwendet werden. Die Nachrichtenstrukturen bestehen wiederum aus Segmenten und diese aus Feldern mit den jeweils zugeordneten Datenelementen.
  - d) Die Datenelemente sind über Datentypen definiert, die ihrerseits aus den entsprechenden Komponenten bestehen.
  - e) Die Datenelemente und die Datentypen beziehen sich wiederum auf Tabellen, die ihrerseits aus Tabellenwerten bestehen.
  - f) In vorhergehender Abbildung ist nur eine Domäne (hier: ADT) symbolisiert. Dies müsste wie in V3 auch für jede einzelne wiederholt werden, was durch den Pfeil nach rechts angedeutet ist.

### 10.6.3. OWL-Elemente für HL7 v2.x

Für eine Umsetzung/Modellierung in OWL werden daher folgende Elemente benötigt, die in nachfolgender Tabelle als Spezialisierung dargestellt sind:

Tabelle 11: Informationselemente von HL7 v2.x und deren Umsetzung in OWL

Information	Umsetzung als
<ul style="list-style-type: none"> <li>• Ereignis</li> <li>• Nachrichtenelemente                             <ul style="list-style-type: none"> <li>○ Segmentgruppen                                     <ul style="list-style-type: none"> <li>▪ Nachricht</li> </ul> </li> <li>○ Segmente</li> <li>○ Datenelemente</li> <li>○ Felder</li> <li>○ Tabellen</li> <li>○ Tabellenwerte</li> </ul> </li> </ul>	Spezialisierung von Klassen in CSO
<ul style="list-style-type: none"> <li>• Attribute                             <ul style="list-style-type: none"> <li>○ Usage/Optionalität                                     <ul style="list-style-type: none"> <li>▪ Werte: R, O, C, RE, CE, X, W, B</li> </ul> </li> <li>○ Kardinalität</li> <li>○ Wiederholbar                                     <ul style="list-style-type: none"> <li>▪ Werte: Y/N</li> </ul> </li> <li>○ Länge</li> </ul> </li> </ul>	Relation zu speziellen Konzepten als Klassen in separater Hierarchie (InformationObject-Qualities) Cardinality Length (DatatypeProperty)



• Relationen	ObjectProperty
○ is-a (Vererbung)	Klassenhierarchie
○ hasComponent (Komponenten)	spezielle Relationen, die jeweils in einer Hierarchie strukturiert sind
○ hasSuccessor (Reihenfolge)	
○ hasPredecessor (Reihenfolge)	
○ hasDatatype (Datentypzuordnung)	
○ hasSegment (via hasComponent?)	
○ hasTable (via hasComponent?)	
○ ...	

In Protégé lassen sich diese Klassen und Relationen wie folgt visualisieren:

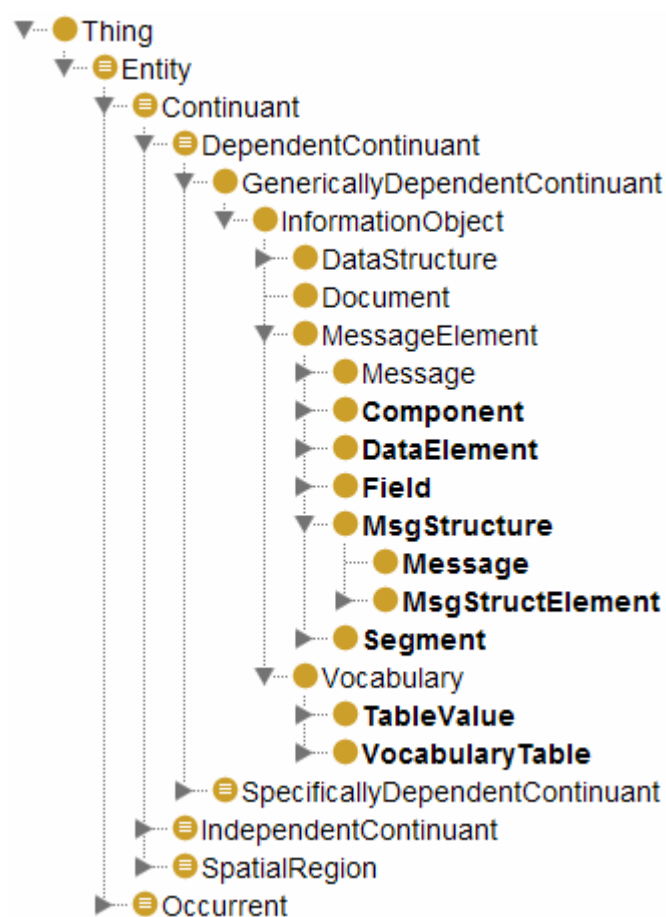


Abbildung 46: Klassenhierarchieübersicht für HL7 v2.6 in Protégé (OWL)

Die Hierarchie der Relationen lässt sich folgendermaßen darstellen:

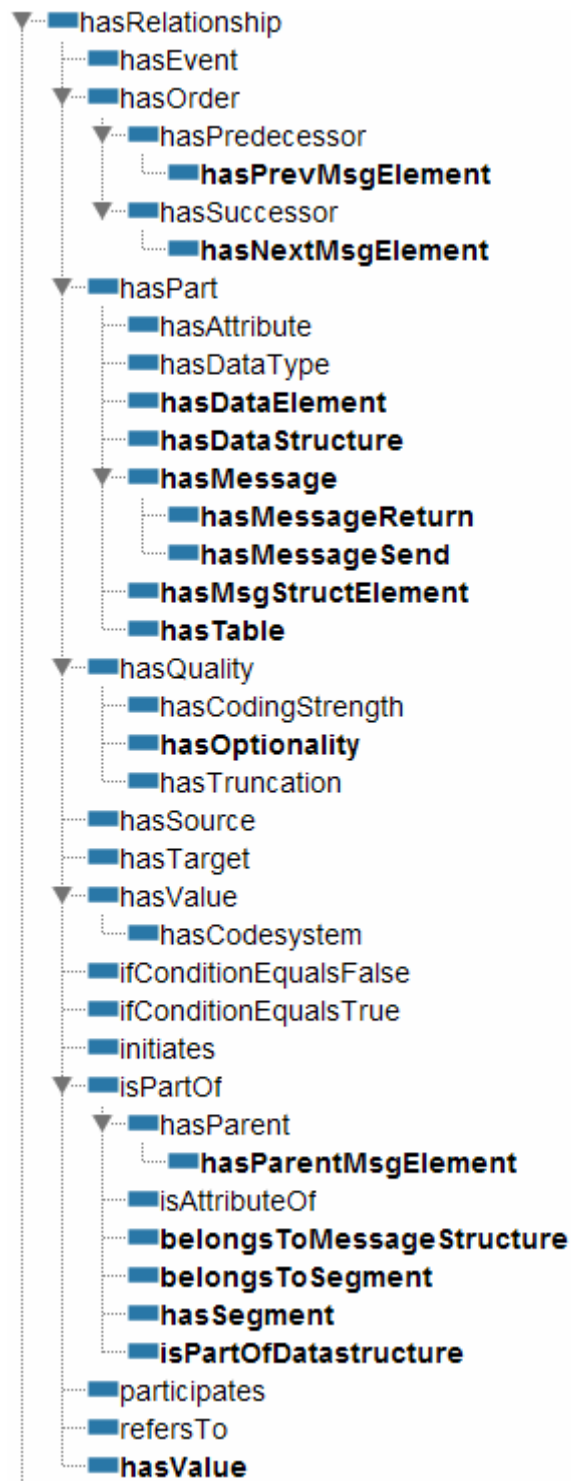
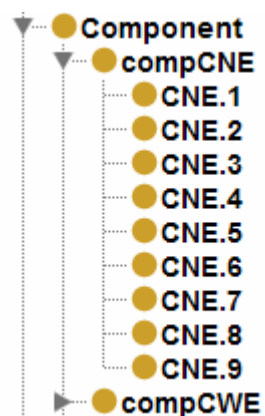


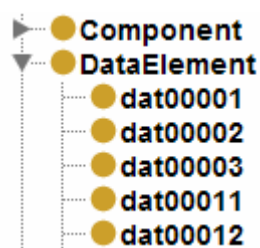
Abbildung 47: Relationshipierarchie für HL7 v2.6 in Protégé (OWL)

Die in der Klassenhierarchie nur ange deuteten Klassen bilden folgende Detailhierarchien, die aber nur ausschnittsweise ange deutet sind:

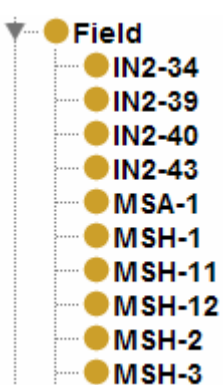
Komponenten



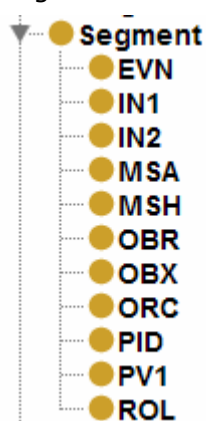
Datenelemente



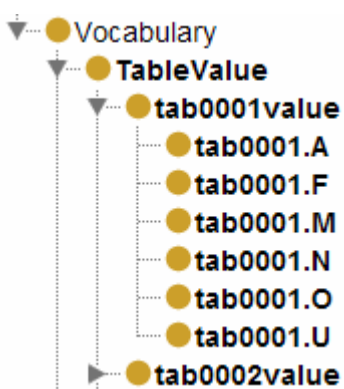
Felder



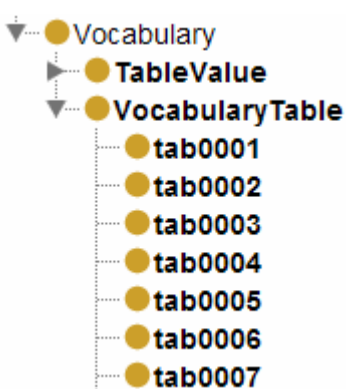
Segmente



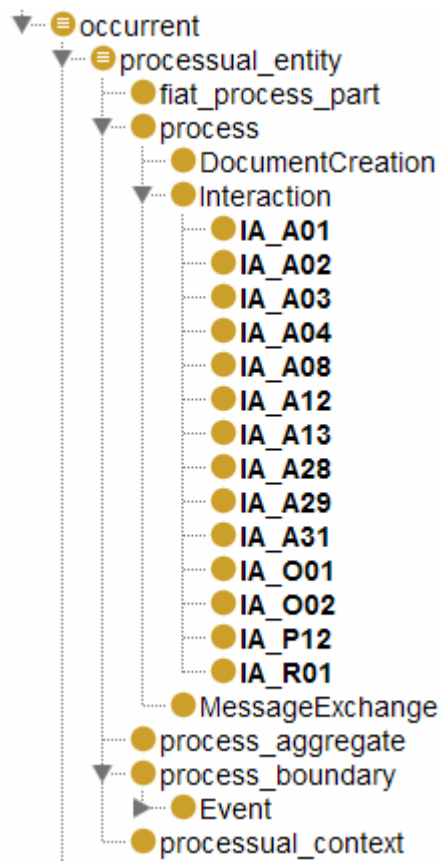
Tabellenwerte



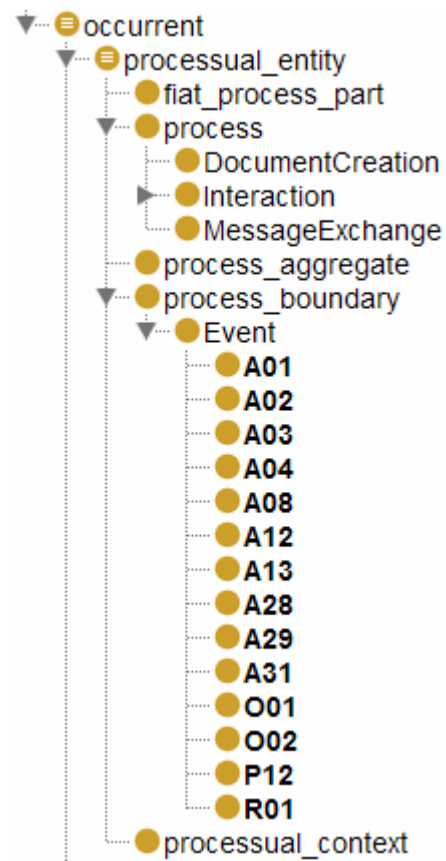
Tabellen



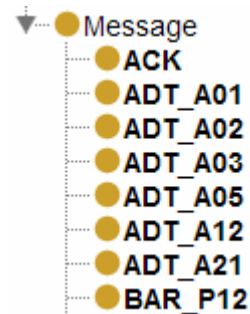
Interaktionen



Ereignisse



Nachrichten



## Nachrichtenstrukturen

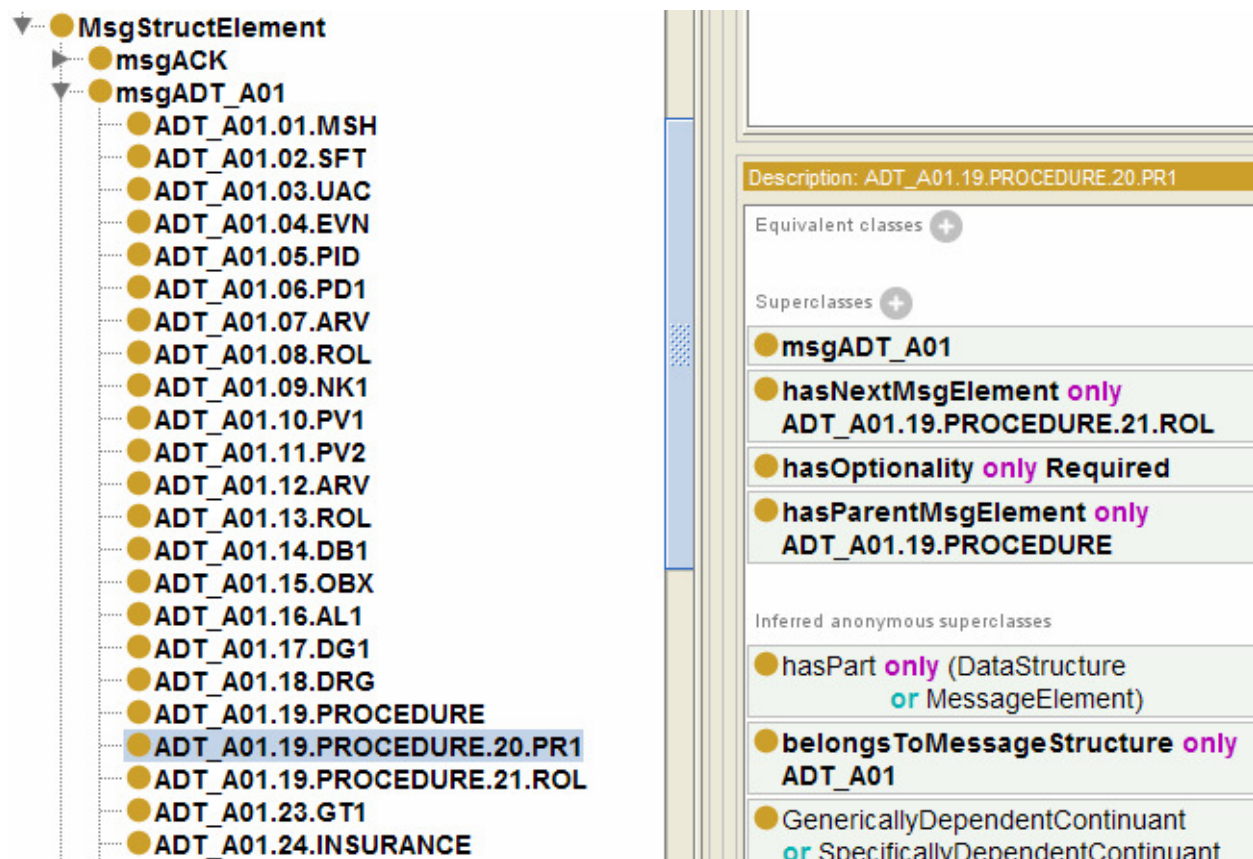


Abbildung 48: Detailklassenhierarchie für HL7 v2.6 in Protégé (OWL)

Die letzte Darstellung gibt einen guten Überblick über den Aufbau der Nachrichtenstrukturen über ObjectProperties: So gehört das markierte Element zur ADT\_A01-Struktur. In dieser gehört es zur Gruppe "PROCEDURE", hat keinen Vorgänger, dafür aber ein Nachfolgeelement („ROL"). An dieser Position ist es „required", muss also in „PROCEDURE" vorkommen. In OWL sieht das formal dann so aus:

```
<owl:Class rdf:ID='ADT_A01.19.PROCEDURE.20.PR1'>
  <!-- 20 -->
  <rdfs:subClassOf>
    <owl:Class rdf:about='#msgADT_A01' />
  </rdfs:subClassOf>
  <!-- Segment relationship hasSegment suppressed! -->
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:FunctionalProperty rdf:about='#hasParentMsgElement' />
      </owl:onProperty>
      <owl:allValuesFrom>
        <owl:Class rdf:about="#ADT_A01.19.PROCEDURE" />
      </owl:allValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

```
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty>
      <owl:FunctionalProperty rdf:about="#hasNextMsgElement"/>
    </owl:onProperty>
    <owl:allValuesFrom>
      <owl:Class rdf:about="#ADT_A01.19.PROCEDURE.21.ROL"/>
    </owl:allValuesFrom>
  </owl:Restriction>
</rdfs:subClassOf>

<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty>
      <owl:FunctionalProperty rdf:about="#hasOptionality"/>
    </owl:onProperty>
    <owl:allValuesFrom>
      <owl:Class rdf:about="#Required"/>
    </owl:allValuesFrom>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
```

Die anderen in Abbildung 48 aufgelisteten Details werden von der übergeordneten Klasse geerbt.

#### 10.6.4. Beispiel-Instanz PID-Segment

Die beispielhafte PID-Segment-Darstellung beschreibt die semantischen Zusammenhänge, die in OWL als Spezialisierung von Klassen in CSO berücksichtigt werden müssen:

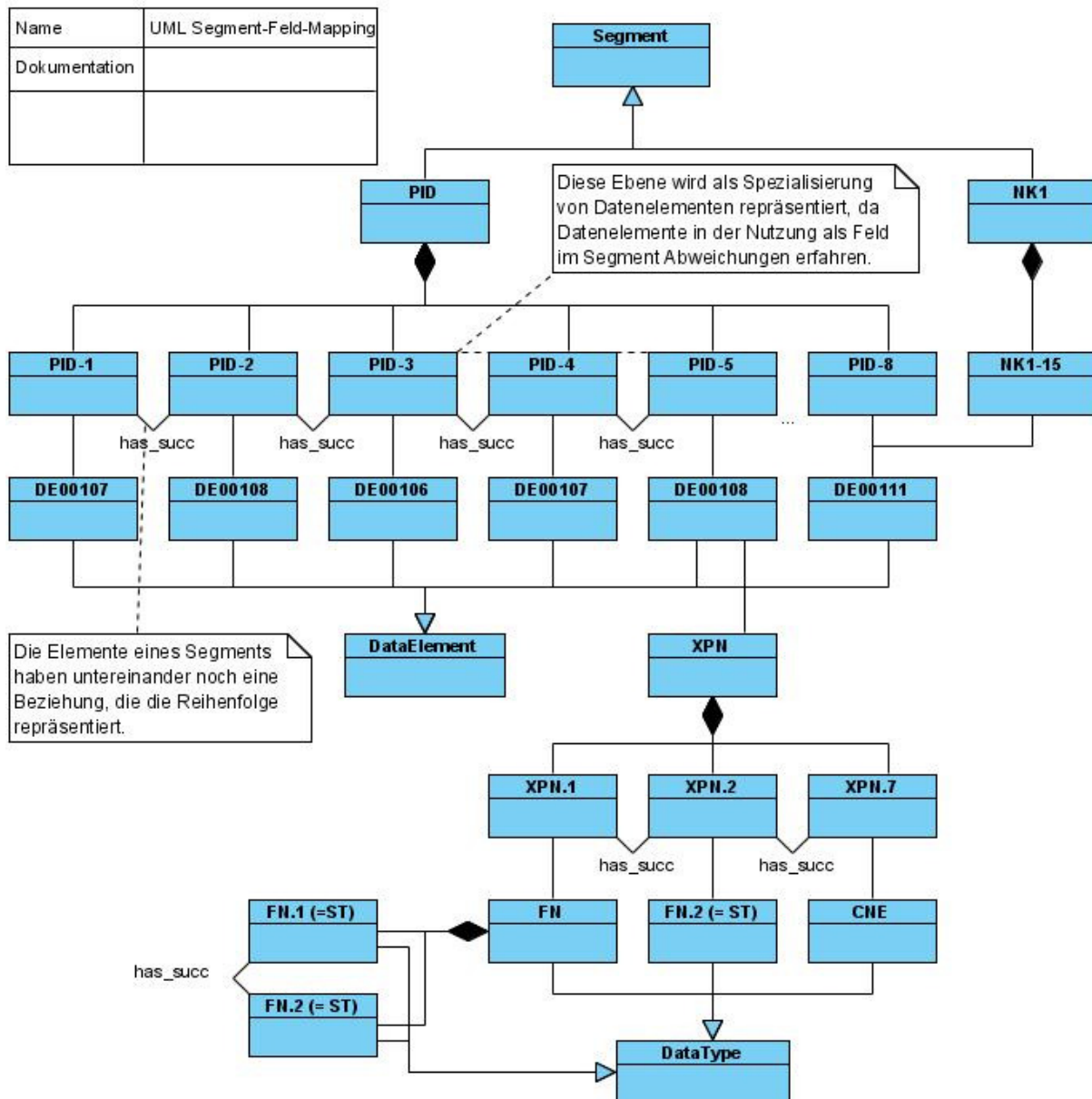


Abbildung 49: Ontologie für HL7 v2.x (UML)

Eine Schwierigkeit in der Umsetzung als "Konzeptualisierung" ist die Entscheidung, ob eine Information in der sog. T-Box oder A-Box (s.o.) untergebracht werden muss, d.h. die Frage, ob es als Mengen (implizites Wissen) oder als Instanz ausgedrückt wird. Da die zu repräsentierenden Informationen und das dazugehörige Wissen ein umfangreiches semantisches Netz darstellen, ist eine Repräsentation in der T-Box – wie bereits in Kap. 7.1.4. T-Box/A-Box dargestellt – angebracht, so dass die Konzepte relativ einfach über ObjectProperty miteinander verknüpft werden können. Dies erleichtert auch die spätere weitere Nutzung – beispielsweise für die Auswertung zur Konfiguration von Kommunikationsservern.

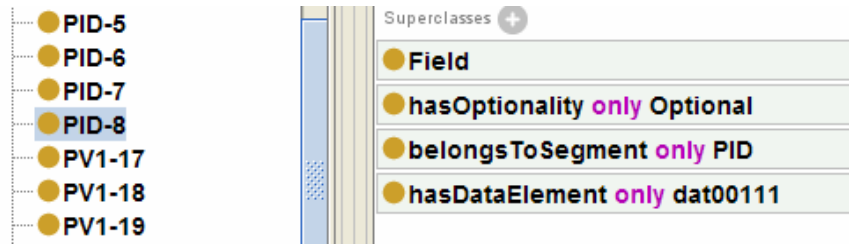


Abbildung 50: Ontologie für HL7 v2.x: PID-8 (Protégé)

In der vorhergehenden Abbildung ist die Referenzierung von PID-8 auf das Datenelement mit der Nummer 00111 in Protégé dargestellt. Für das Feld NK1-15 ergibt sich eine entsprechende Darstellung. Die Auswertung zur Ableitung des Datentyps (hier XPN) muss über die inverse-Relation erfolgen.

Darüber hinaus wird über die Eigenschaften ausgedrückt, dass dieses Feld zum Segment „PID“ gehört („belongsToSegment“, vgl. Abbildung 47) und optional ist.

Nachfolgend die Darstellung in OWL für PID-8 aus vorhergehender Abbildung:

```
<owl:Class rdf:ID='PID-8'>
  <rdfs:subClassOf>
    <owl:Class rdf:about='#Field' />
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:FunctionalProperty rdf:about="#hasDataElement" />
      </owl:onProperty>
      <owl:allValuesFrom>
        <owl:Class rdf:about="#hl7-v2;dat00111" />
      </owl:allValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>

  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:FunctionalProperty rdf:about="#belongsToSegment" />
      </owl:onProperty>
      <owl:allValuesFrom>
        <owl:Class rdf:about="#PID" />
      </owl:allValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>

  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:FunctionalProperty rdf:about="#&cso;hasOptionality" />
      </owl:onProperty>
      <owl:allValuesFrom>
        <owl:Class rdf:about='#Optional' />
      </owl:allValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```



---

## 10.7. Zwischenfazit

Die geforderte "Konzeptualisierung" [Grub1993] muss in der Ontologie in jeweils eigenen Konzepten (OWL-Klassen) resultieren, damit ein Abbildungsprozess möglich ist. Dadurch ergibt sich die zuvor aufgeführte detailreiche Struktur.

## 10.8. Konsistenzregeln

Ein wichtiger Punkt beim Aufbau einer Ontologie ist die Definition von Konsistenzregeln, um feststellen zu können, ob die Ontologie in sich konsistent ist. Diese Regeln werden in OWL über ObjectProperties und DataProperties definiert, die sie auswertbar machen, so dass sie beispielsweise mit einem Reasoner überprüfbar sind. Die bereits vorhergehend erwähnten Zwischenstufen in den Hierarchien unterstützen diese Vorgehensweise.

### 10.8.1. Beispielregeln

Im Folgenden sind einige Beispielregeln aufgeführt:

- Jedes Segment muss mindestens 1 Feld haben.
- Jedes Feld muss genau einem Segment zugeordnet sein.
- Jedes Feld muss genau einem Datenelement zugeordnet sein.
- Ein Datenelement kann einem oder mehreren Feldern zugeordnet sein.
- Jedes Feld hat einen Datentyp. (Ausnahme: "withdrawn elements")
- Jedes Feld muss eine Konformanzklärung haben.
- Jede Komponente hat einen Datentyp.
- Ein Datentyp besteht aus einer oder mehreren Komponente(n).
- Eine Nachricht besteht aus einer Segmentgruppe.
- Eine Segmentgruppe besteht aus einem oder mehreren Segmente(n) oder Segmentgruppe(n).
- ...

Nachfolgend ist ein Beispiel in OWL und Protégé dargestellt, das ausdrückt, dass die Relation „hasDataElement“ zwischen einem Feld und einem Datenelement aufgebaut wird:

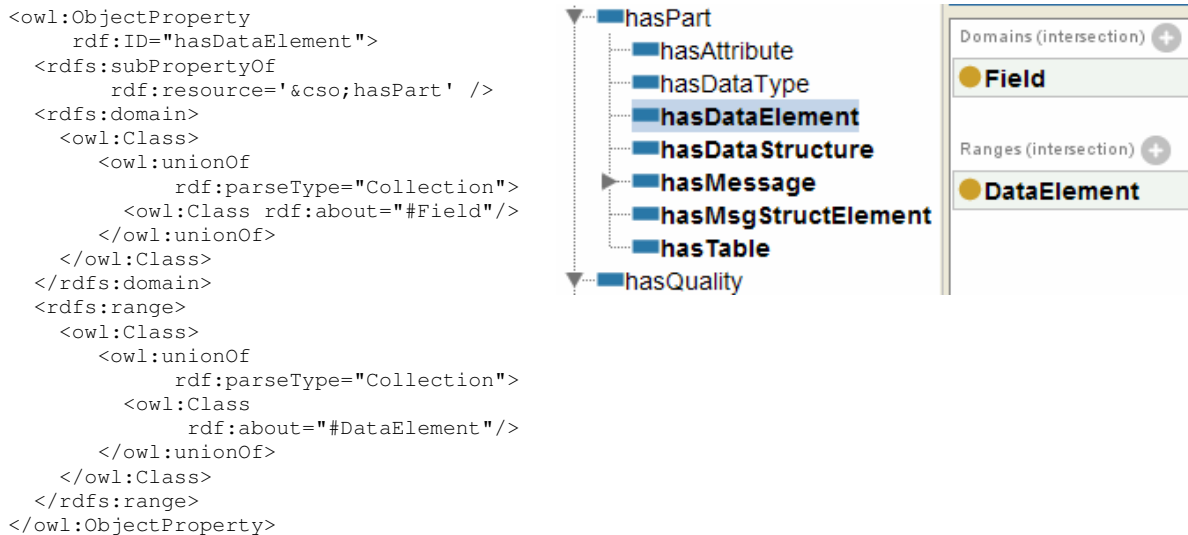


Abbildung 51: Konsistenzregel in OWL und Protégé

Im Gegenzug muss dann jedes Konzept, das eine Spezialisierung von „Field“ ist, genau eine Objektrelation zu einer Spezialisierung von „DataElement“ haben.

### 10.8.2. Umsetzung mittels SWRL

Zur Realisierung der Regeln bietet sich die Semantic Web Rule Language (SWRL) [SWRL] an. Sie erlaubt den Ausdruck komplexerer Sachverhalte, die mehrere Bedingungen miteinander verknüpfen. In dieser Arbeit wird dieser Ansatz aber nicht weiter verfolgt.

## 11. Analyse der Fachdomäne

Alle Informationen, die übertragen werden, entstammen einer Fachdomäne und werden typischerweise in einer Anwendung persistent gespeichert. Ausgenommen davon sind lediglich die reinen Steuerungsinformationen für die Nachrichtenübertragung. Aber auch für diese muss es eine explizite Modellierung in einer – möglicherweise separaten – Fachdomäne geben. Für die Betrachtung der prinzipiellen Umsetzung kann dies aber vernachlässigt werden.

Die verschiedenen Informationen lassen sich als Hierarchie darstellen (vgl. Abbildung 30 und Abbildung 31):

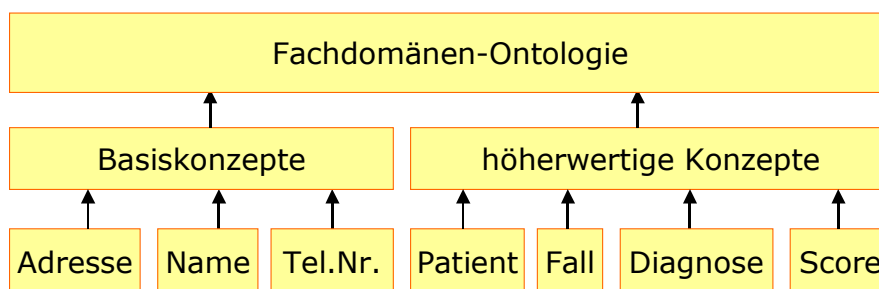


Abbildung 52: Struktur in der Fachdomäne

---

Datentypen können als Basiskonzepte realisiert werden, welche in den höherwertigen Konzepten wiederverwendet werden. Diese Unterscheidung ist allerdings lediglich zur Nachverfolgung interessant. In der weiteren Betrachtung werden alle Konzepte gleichwertig behandelt.

In anderen Fachdomänen-Ontologien wird nicht nur zwischen Basiskonzepten und höherwertigen Konzepten unterschieden, sondern neben den Basic Concepts die Composite Concepts and Meta-Concepts zugelassen (z.B. für Policy-Domänen), wodurch eine hochflexible und umfassende Ontologie-Hierarchie representiert wird [Blob2006].

Nach Blobel [Blob2007d] muss die Architektur umso granularer sein, je unterschiedlicher die zugrunde liegenden Policies und Regelwerke sind, da nur auf dieser Ebene ein Bridging der Policies/Regelwerke vereinbart und ein gemeinsamer Nenner gefunden (Mapping realisiert) werden kann. Nur auf der Ebene gleicher Policies und Regelwerke (Kontext) ist direkte Interoperabilität erreichbar, so dass das System bis auf die gleiche kontextuelle Ebene spezialisiert und auf dieser Ebene die Kommunikation und Kooperation realisiert werden muss. Auf höheren Ebenen ist das Aushandeln gemeinsamer Konzepte (negotiation) unverzichtbar.

Nachfolgend werden einige Beispiele vorgestellt, anhand derer das vorgeschlagene Verfahren überprüft werden soll. Eine unmittelbare Korrektheit in der Repräsentation der Fachdomäne ist für das Ergebnis letztendlich ohne Belang, da die anschließende Nutzung der Ergebnisse in realen Systemen erfolgen soll, die sich aber hinsichtlich der repräsentierten Fachdomäne unterscheiden.

### ***11.1. Beispiel: Name, Adresse und Telefonnummer***

Drei einfache Beispiele sind Namen, Adressen und Telefonnummern. Im Prinzip handelt es sich hier um eine Struktur für einfache textuelle Informationen. Diese Daten werden durch wenige Codes (Typ<sup>34</sup>, Zweck) kontrolliert.

---

<sup>34</sup> An dieser Stelle fällt auf, dass "Name" nicht über eine separate Angabe des Zwecks verfügt. Die Typen "Displayname" oder "Nickname" beschreiben aber eher einen Zweck denn einen separaten Typ.

Name	Adresse	Telefonnummer
-Nachname	-Strasse	-Land
-Vorname	-PLZ	-Ortsvorwahl
-Titel	-Ort	-Anschluß
-Anrede	-Land	
-Typ (legal, Geburtsname, ..)	-Typ (zu Hause, Arbeitsplatz, Ferien, ..)	-Typ (Tel., Fax, ..)
	-Zweck (priv., geschäftl.)	-Zweck (priv., geschäftl.)
-Gültigkeitszeitraum	-Gültigkeitszeitraum	-Gültigkeitszeitraum

Abbildung 53: Name, Adresse und Telefonnummer

Auch wenn diese Informationen als Basiskonzepte jeweils durch einen entsprechenden Datentyp im Kommunikationsstandard ausgedrückt werden, so sind die sich dahinter verbergenden Informationen komplizierterer Natur. In einer Ontologie müssen sie durch mehrere separate Konzepte realisiert werden, die zueinander in einer entsprechenden Beziehung stehen. Notfalls müssen dazu sogar individuelle explizite "Beziehungskonzepte" eingeführt werden. Die vorangehend aufgelisteten Konzepte stellen gemäß BFO eine Spezialisierung eines Identifiers dar, wie die nachfolgende Tabelle zeigt. Nicht genutzte Klassen aus ACGT sind dabei durchgestrichen:

<ul style="list-style-type: none"> <li>• Identifier (acgt)               <ul style="list-style-type: none"> <li>○ Name (acgt)                   <ul style="list-style-type: none"> <li>▪ ArtistName (acgt-fo)</li> <li>▪ DisplayName (acgt-fo)</li> <li>▪ LegalName (acgt-fo)</li> <li>▪ BirthName (acgt-fo)</li> <li>▪ NameAtBirth (acgt-fo)</li> <li>▪ Nickname(acgt-fo)</li> <li><del>▪ FirstName (acgt)</del></li> <li><del>▪ FullName (acgt)</del></li> <li><del>▪ LastName (acgt)</del> <ul style="list-style-type: none"> <li><del>• MaidenName (acgt)</del></li> </ul> </li> </ul> </li> <li>○ NamePart (acgt-fo)                   <ul style="list-style-type: none"> <li>▪ FirstName (acgt-fo)</li> <li>▪ GivenName (acgt-fo)</li> <li>▪ MiddleInitials (acgt-fo)</li> <li>▪ NamePrefix (acgt-fo)</li> <li>▪ NameSuffix (acgt-fo)</li> <li>▪ Title/Anrede (acgt-fo)</li> </ul> </li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>• Identifier (acgt)               <ul style="list-style-type: none"> <li>○ Contact                   <ul style="list-style-type: none"> <li>▪ eMail</li> <li>▪ TelephoneNumber (acgt)                       <ul style="list-style-type: none"> <li>• SpeedDialCode (acgt-fo)</li> </ul> </li> </ul> </li> <li>○ ContactPart                   <ul style="list-style-type: none"> <li>▪ TelephoneNumberPart(acgt)                       <ul style="list-style-type: none"> <li>• CityCode (acgt-fo)</li> <li>• CountryCode (acgt-fo)</li> <li>• ExtensionNumber (acgt-fo)</li> <li>• LocalCode (acgt-fo)</li> </ul> </li> </ul> </li> </ul> </li> </ul>

- Identifier (acgt)
  - Address<sup>35</sup> (acgt)
    - AddressOfWorkingPlace (acgt-fo)
    - HomeAddress (acgt-fo)
    - VacationHome (acgt-fo)
  - AddressPart<sup>36</sup> (acgt)
    - CountryName (acgt)
      - BirthCountryName (acgt)
    - HouseNumber (acgt)
    - PlaceName (acgt)
      - CityName (acgt)
        - BirthCityName (acgt)
    - PostalCode (acgt)
    - StreetName (acgt)
- IdentifierQuality
  - NameRepresentation (acgt-fo)
    - PhoneticNameRepresentation (acgt-fo)
    - AlphabeticNameRepresentation (acgt-fo)
    - IdeographicNameRepresentation (acgt-fo)
  - IdentifierUse (acgt-fo)
    - Business (acgt-fo)
      - WorkingPlace (acgt-fo)
    - Private (acgt-fo)
      - PrivateVacation (acgt-fo)
  - PhoneEquipmentType (acgt-fo)
    - Tel (acgt-fo)
    - Fax (acgt-fo)
  - IdentifierUse (acgt-fo)
    - Private (acgt-fo)
      - PrivateVacation (acgt-fo)
    - Business (acgt-fo)

Abbildung 54: Ontologische Konzepte für Namen, Adressen und Telefonnummern

Anm.: Die für diese Dissertation notwendigen Konzepte sollten als Spezialisierung der bereits vorhandenen Konzepte in ACGT aufgenommen werden. Dies führt dann konsequenterweise zur Diskussion, ob die bereits vorhandenen Konzepte korrekt definiert worden sind oder nicht. Da dies ein langwieriger und zu konsentierender Prozess ist, soll im Weiteren darauf nicht eingegangen werden. Der Einfachheit halber werden die benötigten Konzepte „so gut wie möglich“ in der vorhandenen Hierarchie einsortiert.

In allen Fällen handelt es sich um Identifikatoren, über die bedingt Personen, Plätze oder Kontaktdaten identifiziert werden können. In der Regel muss der entsprechende Identifikator weiter spezialisiert werden. So werden beispielsweise die Namen weiter

<sup>35</sup> An dieser Stelle werden die verschiedenen Nutzungsmöglichkeiten als Spezialisierungen ausmodelliert. Es wäre genauso gut möglich, diese über den IdentifierUse weiter zu qualifizieren.

<sup>36</sup> Bei „AddressPart“ und anderen Teilhierarchien stellt sich die Frage, ob diese nicht besser als Qualifier repräsentiert werden. Diese Diskussion muss aber bei ACGT geführt werden. Für diese Arbeit wird auf die bestehende Hierarchie zurückgegriffen.

unterteilt in rechtsverbindliche Namen (gemäß Personalausweis), angezeigte Namen (beispielsweise für Briefe) oder Geburtsnamen<sup>37</sup>.

Im asiatischen Sprachraum können die spezialisierten Namen jeweils noch weiter in die alphabetische, ideographische und phonetische Darstellung unterteilt werden. Da es sich hier um eine orthogonale weitere Untergliederung handelt, kann diese alternativ auch über eine weitere Hierarchie zum Ausdruck gebracht werden. So können beispielsweise die Spezialisierungen der "PhoneNumber" als "defined class" umgesetzt werden. Welche Variante zu favorisieren ist, hängt vom Anwendungsfall ab. Gemäß Gruber und Westenthaler [GruWes2006] gibt es ja nicht die „eine richtige“ Ontologie. Die zu nutzende Domänenontologie hängt folglich von zusätzlichen Faktoren (environmental conditions, rules, relations etc.) ab.

Auf der anderen Seite besteht ein solcher Identifikator aus entsprechenden Teilen, die geeignet gekennzeichnet sind – in vorhergehender Abbildung mit "...Part" bezeichnet.

Weder in BFO noch in ACGT wird diesem Umstand entsprechend Rechnung getragen. In dieser Arbeit führt eine separate, auf ACGT aufbauende Domänen-Ontologie diese Konzepte ein. Damit wird dem AAA-Prinzip [AllHen2008] Rechnung getragen und eine einfache Identifikation der neuen Konzepte für eine Diskussion mit den Autoren von ACGT ermöglicht.

## **11.2. Beispiel: "Fall"**

Ein weiteres Beispiel ist das Konzept des "Falles". Darunter kann sowohl ein administrativer, abrechnungstechnischer als auch ein medizinischer Fall verstanden werden. Ein Fall ist mit mehreren Personen verbunden, die den Patienten, Ärzte oder Pfleger darstellen. Genaugenommen sind das Rollen, die von unterschiedlichen Personen gespielt werden ("Patient", "Arzt"), und keine Spezialisierung einer Person.

---

<sup>37</sup> In Deutschland bezieht sich das Konzept des Geburtsnamens ausschließlich auf den Nachnamen. In anderen Ländern muss dies nicht notwendigerweise ebenso gehandhabt werden. Auch ist eine Unterscheidung in Geburtsname und Mädchenname nur schwer zu treffen.

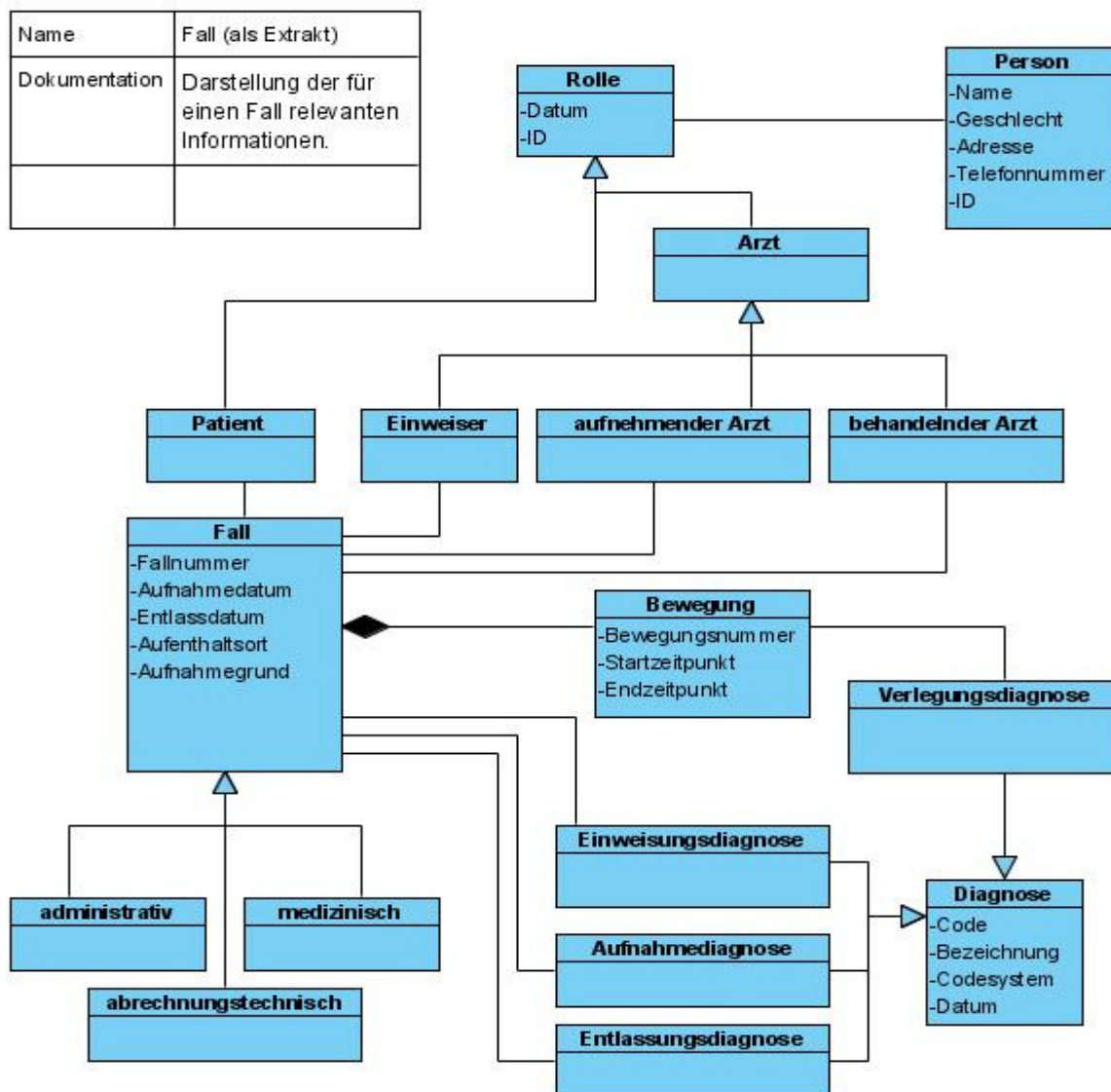


Abbildung 55: UML-Modell für Fall (als Extrakt)

Die vorhergehende Abbildung stellt nur beispielhaft einen Ausschnitt der relevanten Daten dar.

Auch diese Informationen sind in einer Ontologie über Klassen und Relationen entsprechend darzustellen. Umgekehrt kann (muß) eine Ontologie dazu benutzt werden, Zweideutigkeiten in den Nachrichten zu überwinden.

### 11.3. Beispiel: Diagnosen

In den Nachrichtenprofilen für HL7 Version 2.5 [HL7-DE 2007] sowie im Diagnoseleitfaden [HL7 Diag] sind die Basisdaten für Diagnosen entsprechend zusammengestellt.

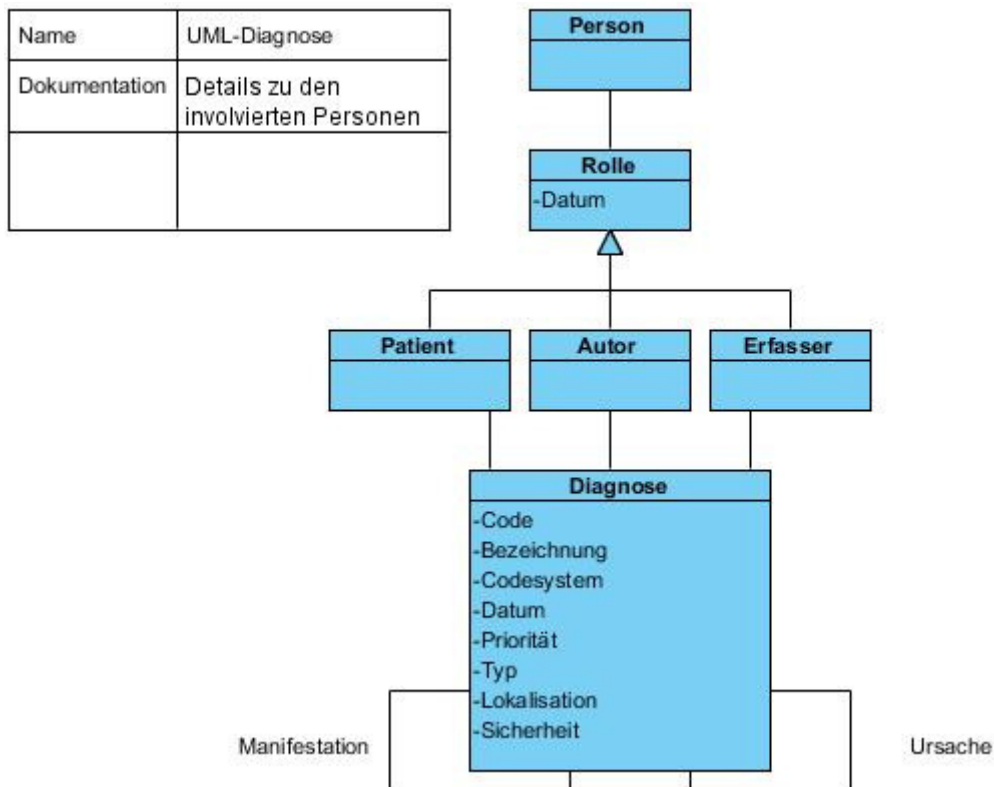


Abbildung 56: Diagnose

Mitunter stehen Diagnosen mit anderen Diagnosen in Beziehung. Dies ist beispielsweise bei der sog. Kreuz-/Stern-Notation (Ursache und Manifestation) für ICD 10 der Fall [ICD, ICD10GM2009].

Der HL7 V3 Diagnoseleitfaden [HL7-Diag] für Tumordiagnosen realisiert bereits kompliziertere Modelle.

#### **11.4. Beispiel: Scores und Assessments**

Bei den Scores und Assessments handelt es sich um Bewertungsskalen, die Einzelinformationen gruppieren und durch einfach interpretierbare Zahlen darstellen. Das nachfolgende Modell [OemBlo2007a] stellt ein HL7 Template dar, das als Spezialisierung des sog. Clinical Statement Patterns entwickelt worden ist [OemThu2007b]. Es entstammt dem entsprechenden Implementierungsleitfaden [OemThu2007a].



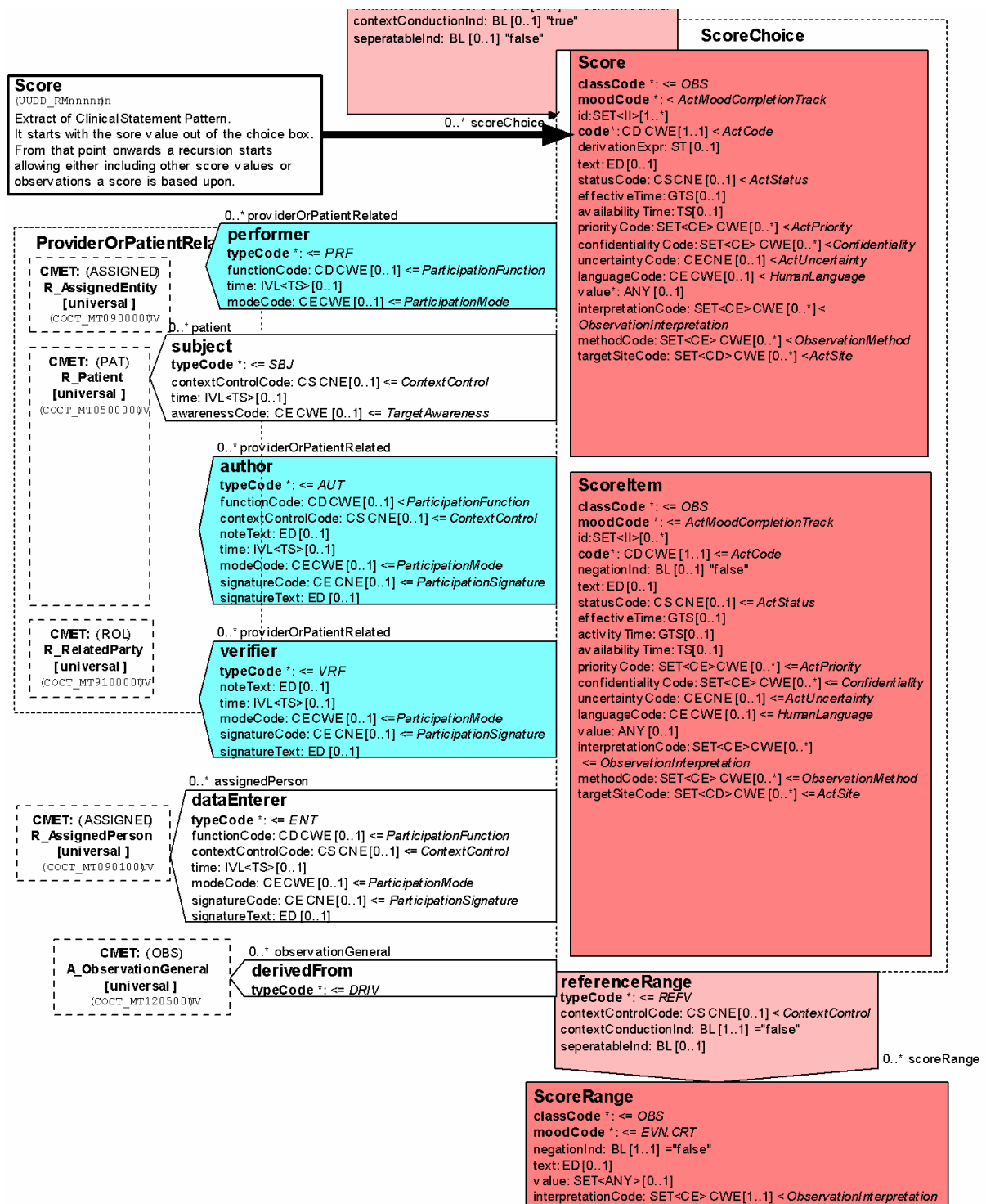


Abbildung 57: HL7 V3 Modell für Scores

Das vorangehende Analysemodell wurde im Rahmen der Arbeit auch in eine Datenbank mit einem vereinfachten Modell überführt, in der alle für eine Kommunikation relevanten Daten (Werte, Referenzbereiche, Interpretationen sowie die Codes in verschiedenen Codesystemen) verwaltet werden. In dieser Entwicklungsprozesssicht sind daher keine Felder zum Speichern der Ergebniswerte enthalten. Dies bliebe einer vollständigen Betrachtung der Entwicklungsprozessdimension des GCM im Rahmen der konkreten Spezifikation einer Anwendung vorbehalten.

Der wichtigste Aspekt ist die Tatsache, dass ein Score einen Befund gemäß HL7 V3 darstellt. Daher wird ein Score als Spezialisierung eines Befundes modelliert (siehe Kommentar a. in der nachfolgenden Abbildung). In der erwähnten Datenbank werden diese beiden Klassen in einer gemeinsamen Tabelle verwaltet. Das vereinfacht die Handhabung der Rekursion, d.h. den Fall, dass ein Befund wiederum andere Befunde enthalten kann – oder wie hier ein Score einen anderen Score enthält. Als Konsequenz stellt jeder Eintrag somit letztendlich ein eigenes Konzept dar, das in verschiedenen Definitionen genutzt werden kann.

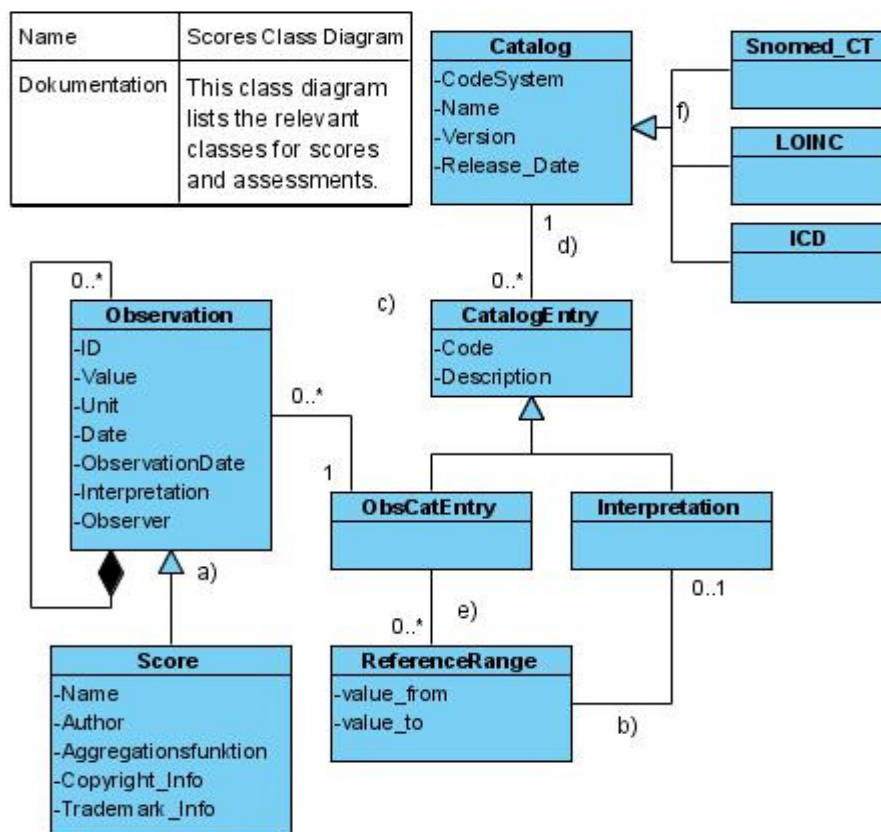


Abbildung 58: UML-Modell für Scores

Ein weiterer wichtiger Punkt ist die Nutzung von Codes zur Repräsentation semantischer Konzepte. In Abbildung 58 repräsentiert die Klasse CatalogEntry ein Konzept mit einem Code und einer Beschreibung aus einem Codesystem (d). Das Konzept kann aus einem Katalog wie LOINC [LOINC] oder Snomed CT [Snomed CT] stammen. Die zwei Spezialisierungen als ObsCatEntry und Interpretation sollen die Referenzierungen erleichtern. Die erste Klasse identifiziert Konzepte für Befunde, die zweite für Interpretationen.

Jedes Konzept für einen Befund kann mit einem Referenzbereich verknüpft werden (e) in obiger Abbildung), der die erwarteten (gültigen) Werte beschreibt. Jeder einzelne Bereich ist wiederum mit einer Interpretation verknüpft (b) in obiger Abb.).

Ein Score hat im Gegensatz zu einfachen Beobachtungen/Messungen typischerweise mehrere Referenzbereiche mit verschiedenen Interpretationen.

### 11.5. Einsatz der BFO (Basic Formal Ontology) und ACGT (Advancing Clinico Genomic Trials on Cancer)

IFOMIS (Institute for Formal Ontology and Medical Information Science, [IFOMIS]), eine Einrichtung an der Universität des Saarlandes, hat an der Entwicklung von 2 Ontologien mitgewirkt: BFO und ACGT. ACGT, die "Advancing Clinico Genomic Trials on Cancer" Ontologie baut auf BFO, der "Basic Formal Ontology" auf. Die grundlegende Struktur ist ausschnittsweise in der nachfolgenden Abbildung aufgelistet, wobei der Namespace der jeweiligen Ontologie<sup>38</sup> in Klammern angegeben ist. Soweit notwendig werden in einer separaten Spalte zusätzliche Erläuterungen aufgeführt.

Tabelle 12: BFO Modell/Hierarchie

Element (Klasse)	Erläuterung
Thing entity	
continuant dependent_continuant <b>generically_dependent_continuant</b> specifically_dependent_continuant <b>quality</b> realizable_entity disposition function <b>role</b> independent_continuant material_entity fiat_object_part <b>object</b> object_aggregate object_boundary site spatial_region	
occurent processual_entity fiat_process_part <b>process</b> process_aggregate <b>process_boundary</b>	

<sup>38</sup> Genaugenommen ist BFO in mehrere Einzelontologien (bfo, snap, span) unterteilt (Stand 25.2.10). Da dieses Detail für diese Arbeit keine Rolle spielt, wird hier zur Vereinfachung alles unter „bfo“ subsumiert. Alle aufgelisteten Klassen einschließlich „entity“ gehören zu BFO.

Element (Klasse)	Erläuterung
processual_context (bfo) spatiotemporal_region (bfo) temporal_region (span)	

ACGT stellt wie bereits erwähnt eine Erweiterung von BFO dar, wobei die weiter zu spezialisierenden Konzepte in der vorhergehenden Darstellung fett markiert sind.

Wie ebenfalls bereits angedeutet, muss die durch ACGT bereitgestellte Fachdomäne erweitert werden. Diese Erweiterungen sind in den nachfolgenden Tabellen farblich markiert und zum Teil erläutert sowie durch das Kürzel „acgt-fo“ markiert.

Das InformationObject von ACGT ist eine Spezialisierung des „generically dependent continuant“. Die eigenen Erweiterungen („acgt-fo“) sind in der Spalte mit den Erläuterungen durch eine Hintergrundfarbe markiert:

Tabelle 13: ACGT + BFO Modell/Hierarchie (generically\_dependent\_continuant)

Element (Klasse)	Erläuterung
generically_dependent_continuant (snap)	
InformationObject (acgt) Design (acgt) Document (acgt)	
Assessment (acgt-fo)	
Identifier (acgt)	
AccountNumber (acgt-fo) CreditCardNumber (acgt-fo)	stellvertretend für weitere Nummern
EncounterIdentifier (acgt-fo) MovementIdentifier (acgt-fo) VisitIdentifier (acgt-fo) InpatientVisitIdentifier (acgt-fo) OutpatientVisitIdentifier (acgt-fo)	Identifier für die verschiedenen Abschnitte verschiedener Aufenthalte
NamePart <sup>39</sup> (acgt-fo) FirstName (acgt-fo) GivenName (acgt-fo) MiddleInitials (acgt-fo) NamePrefix (acgt-fo) NameSuffix (acgt-fo) Title (acgt-fo)	anstelle der Spezialisierungen aus ACGT (weiter unten durchgestrichen dargestellt)
PersonIdentifier (acgt-fo)	
TelephoneNumberPart (acgt-fo) CityCode (acgt-fo) CountryCode (acgt-fo) ExtensionNumber (acgt-fo)	zwecks Zuordnung der Details

<sup>39</sup> Wie bereits bei ACGT angeführt stellt sich hier die Frage, ob dies eine Spezialisierung von Identifier ist oder nicht.

Element (Klasse)	Erläuterung
LocalCode (acgt-fo) SpeedDial (acgt-fo)	
Address (acgt)	
AddressOfWorkingPlace (acgt-fo) HomeAddress (acgt-fo) VacationAddress (Acgt-fo)	repräsentiert die verschiedenen Nutzungsarten
AddressPart (acgt) CountryName (acgt) BirthCountryName (acgt) HouseNumber (acgt) PlaceName (acgt) CityName (acgt) BirthCityName (acgt) PostalCode (acgt) StreetName (acgt) CRFNumber (acgt) ... InstitutionIdentifier (acgt) HospitalIdentifier (acgt)	
LegalGuardianIdentifier (acgt) MedicalExpertIdentifier (acgt)	genaugenommen wird hier die Rolle identifiziert, d.h. es müsste einen "RoleIdentifier" geben
Name (acgt)	
ArtistName (acgt-fo) DisplayName (acgt-fo) LegalName (acgt-fo) NameAtBirth (acgt-fo) NickName (acgt-fo)	verschiedene Spezialisierungen von Namen
<del>FirstName (acgt)</del> <del>FullName (acgt)</del> <del>LastName (acgt)</del> <del>MaidenName (acgt)</del>	werden nicht benötigt, da Strukturkonflikt
PatientIdentifier (acgt) ...	Vgl. Guardian-ID
RegistryIdentifier (acgt) TelephoneNumber (acgt) Proposition (acgt) Publication (acgt)	

Des Weiteren wurden beispielhaft ein paar weitere „Qualitäten“ definiert:

Tabelle 14: ACGT + BFO Modell/Hierarchie (quality)

Element (Klasse)	Erläuterung
quality (snap)	
MaritalStatus (acgt-fo) Divorced (acgt-fo) Married (acgt-fo) Single (acgt-fo)	Beispiele für verschiedene Vokabularien; die jeweiligen Spezialisierungen sind nicht vollständig

Entwicklung einer ontologiebasierten Architektur zur Sicherung semantischer Interoperabilität zwischen Kommunikationsstandards im Gesundheitswesen

Element (Klasse)	Erläuterung
Widowed (acgt-fo) Religion (acgt-fo) GreekOrthodox (acgt-fo) Islam (acgt-fo) Jewish (acgt-fo) Protestant (acgt-fo) RomanCatholic (acgt-fo)	
Disease (acgt) FunctionalOrStructural (acgt) Gender (acgt) AmbiguousGender (acgt) Female (acgt) Male (acgt)  Laterality (acgt) Magnitude (acgt) Age (acgt) ...	Wie in HL7 auch besteht hier ein „Konflikt“ zwischen administrativer und genetic gender.

Die Rollen wurden ebenso ergänzt:

Tabelle 15: ACGT + BFO Modell/Hierarchie (role)

Element (Klasse)	Erläuterung
role (snap)	
AdministrativeRole (acgt) ... ProfessionalRole (acgt)	
MedicalAssistant (acgt-fo) MTLA (acgt-fo) MTRA (acgt-fo)	
Jurist ... MedicalPractitioner (acgt) ... Radiologist (acgt)	
LocalRadiologist (acgt) ReferenceRadiologist (acgt)	Beispiele für "unsaubere" Stellen <sup>40</sup> in ACGT.
Surgeon (acgt) ... Nurse (acgt) ... Relative (acgt)	
Child (acgt-fo) Daughter (acgt-fo) Son (acgt-fo)	könnte auch als Inverse-Relation zu Parent etc. definiert werden.

<sup>40</sup> Hier werden verschiedene Aspekte miteinander vermischt, die besser über Properties ausgedrückt werden.

Element (Klasse)	Erläuterung
Offspring (acgt) Parent (acgt)	
Father (acgt-fo)	
Mother (acgt) Progenitor (acgt) Sibling (acgt) Twin (acgt) Therapy (acgt)	

In der letzten Übersicht sind Erweiterungen für Prozesse und Institutionen direkt mit aufgelistet.

Tabelle 16: ACGT + BFO Modell/Hierarchie (independent\_continuant)

Element (Klasse)	Erläuterung
Thing	
entity (bfo) continuant (bfo)	
independent_continuant (bfo) material_entity (bfo) fiat_object_part (bfo) object (bfo)	
Institution (acgt)	
Facility (acgt-fo) Ward (acgt-fo)	
Company (acgt) ... Hospital (acgt) ...	
object_aggregate (bfo) object_boundary (bfo) site (bfo) spatial_region (bfo) occurent (bfo) processual_entity (bfo)	
fiat_process_part (bfo) process (bfo)	
Encounter (acgt-fo) InpatientEncounter (acgt-fo) OutpatientEncounter (acgt-fo)	
AdministrativeProcess (acgt) ClinicalDocumentation (acgt) ... ClinicalResearch (acgt) Consent (acgt) PreClinicalResearch (acgt) ProtocolChange (acgt) Registration (acgt)	

Entwicklung einer ontologiebasierten Architektur zur Sicherung semantischer Interoperabilität zwischen Kommunikationsstandards im Gesundheitswesen

Element (Klasse)	Erläuterung
Admission (acgt-fo) Discharge (acgt-fo) LeaveOfAbsence (acgt-fo) RegisterPerson (acgt-fo) ReturnLeaveOfAbsence (acgt-fo) Transfer (acgt-fo) UpdatePerson (acgt-fo)	
MedicalProcess (acgt) DiagnosticProcess (acgt) ... TherapeuticProcess (acgt) ... MolecularBiologyTechnique (acgt) OrganismalProcess (acgt) ScientificProcess (acgt)	
process_aggregate (bfo) process_boundary (bfo)	
ProcessEnd (acgt) Death (acgt) MeasuringProcessEnd (acgt) ... RecordingProcessEnd (acgt) ... ProcessStart (acgt) Birth (acgt) MeasuringProcessStart (acgt) ... RecordingProcessStart (acgt) ...	
process_result ScientificObservation ScoreResultPart ScoreResult Interpretation ReferenceRange	Für Scores
processual_context (bfo) spatiotemporal_region (bfo) temporal_region (bfo)	

Wie die vorangehende Auflistung zeigt, ist ACGT basierend auf BFO als Referenzmodell zur Modellierung der Fachdomäne im Prinzip tauglich.

Dennoch erscheint ihr Aufbau in einigen Teilen inkonsistent und für unseren Einsatz nicht geeignet: So erfolgt die Spezifikation für „Name“ grundlegend anders als für „Adresse“. Diese Informationen werden deshalb nicht verwendet und sind in der Auflistung des InformationObjects durchgestrichen dargestellt.

Derartige Schwierigkeiten stellen aber keinen Grund dar, von diesen beiden Ontologien Abstand zu nehmen und stattdessen eine neue Ontologie zu entwickeln. Der grundlegende Ansatz in der Nutzung von Ontologien ist die Wiederverwendung bereits definierter Artefakte in Kombination mit eigenen Zusatzinformationen (Annotationen). So



können BFO und ACGT in der bereits veröffentlichten Fassung genutzt werden. Wenn keine Änderungen ausgehandelt werden können, sollten aber darauf aufbauend weitere Ontologien festgelegt werden, die die Nachteile ausgleichen. So werden die benötigten Spezialisierungen für „Name“ (in obiger Darstellung entsprechend annotiert) in Analogie zu „Adresse“ in einer vergleichbaren Notation vorgestellt.

## **12. Ontologien für die Fachdomäne**

Wie bereits angedeutet benötigt ACGT noch Ergänzungen an diversen Stellen in der Spezialisierungshierarchie. Diese sind in der Darstellung in Kap. „11.5. Einsatz der BFO (Basic Formal Ontology) und ACGT (Advancing Clinico Genomic Trials on Cancer) bereits mit eingearbeitet und durch Kommentare erläutert. Nachfolgend wird dies noch an einzelnen Beispielen weiter ausgeführt.

### ***12.1. Beispiel Name/Adresse***

Die verschiedenen Konzepte für Namen und auch Adressen werden über eine Reihe von Relationen realisiert. Dies wird nachfolgend beim Mapping ausführlicher an einem Beispiel erläutert.

### ***12.2. Beispiel: Fall***

ACGT fehlen diverse Konzepte, um über deren ObjectProperties ein Mapping von Kommunikationsstandards im administrativen Bereich durchführen zu können. Diese werden daher unter „Process“ eingeführt:

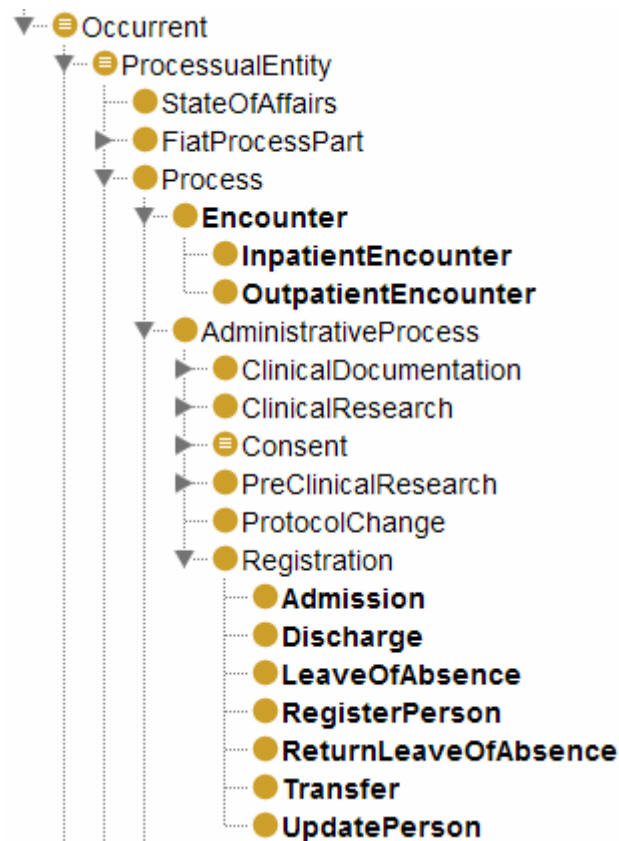


Abbildung 59: Encounter and Registration in ACGT-FO

### 12.3. Beispiel: Scores

Um wissensbasierte Plausibilitätsprüfungen durchführen zu können, muss das UML-Modell (Abbildung 58) in eine OWL-Darstellung konvertiert werden.

#### 12.3.1. UML-OWL-Modell

Im Prinzip ist die Modellierung in UML und OWL sehr ähnlich. Der größte Unterschied ist die Ableitung von einer gemeinsamen Klasse, die owl:Thing genannt wird. Des Weiteren werden einzelne Attribute als Beziehungen zu anderen Klassen realisiert. Eine Person als Spezialisierung eines LivingSubject in der Funktion (Rolle) eines Beobachters oder Autors ist dafür ein gutes Beispiel.

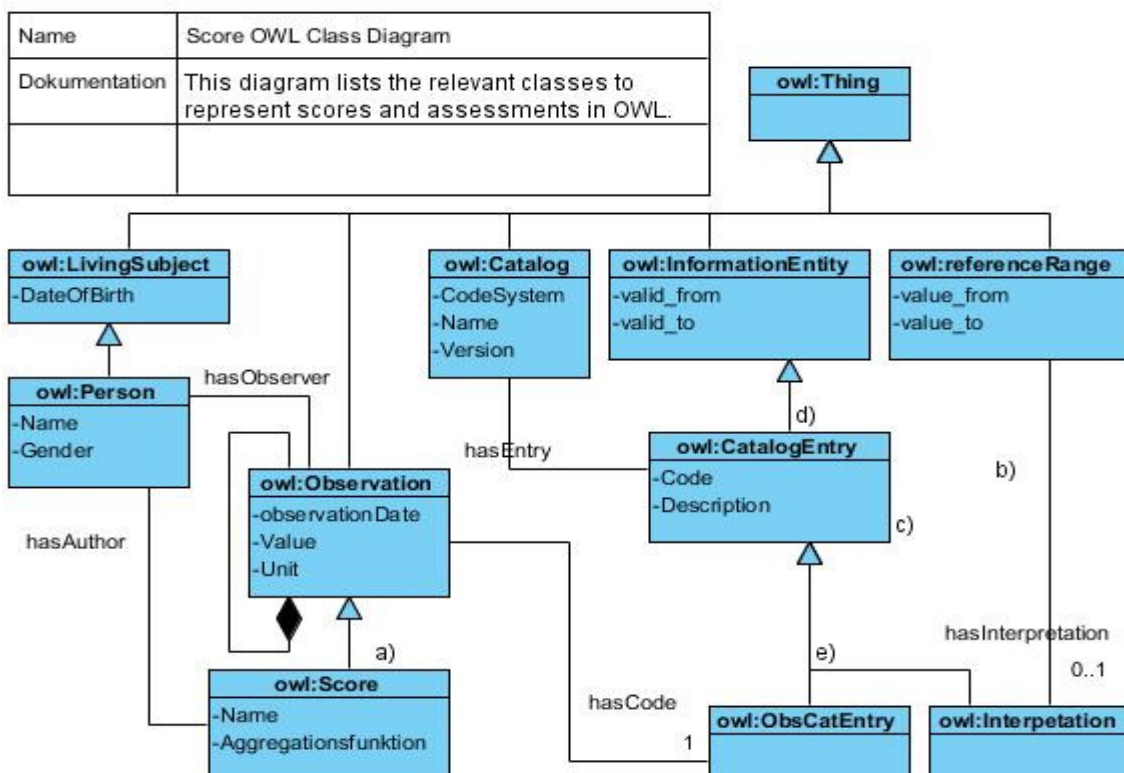


Abbildung 60: UML OWL Modell

Anm.: Die Kommentare aus Abbildung 58 gelten auch für die obige Abbildung.  
Eine vereinfachte direkte erste Umsetzung in Protégé sieht dann wie folgt aus:

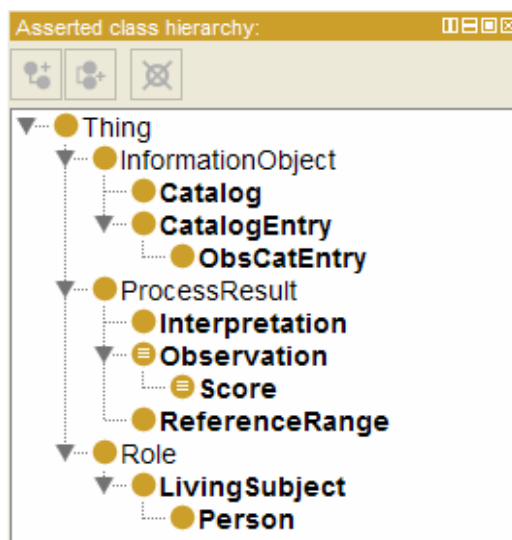


Abbildung 61: Protégé OWL Modell (als direkte Umsetzung)



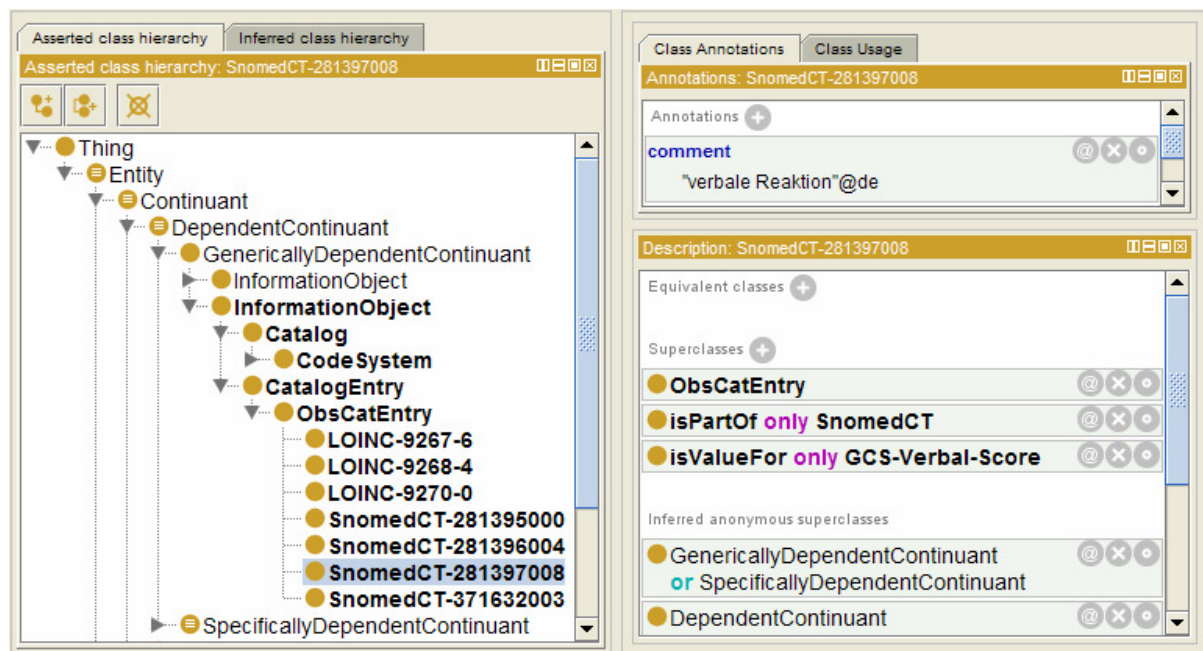


Abbildung 63: Code-Definition für die Scores auf Basis von ACGT

Gleiches gilt für die Interpretationen und Referenzbereiche.

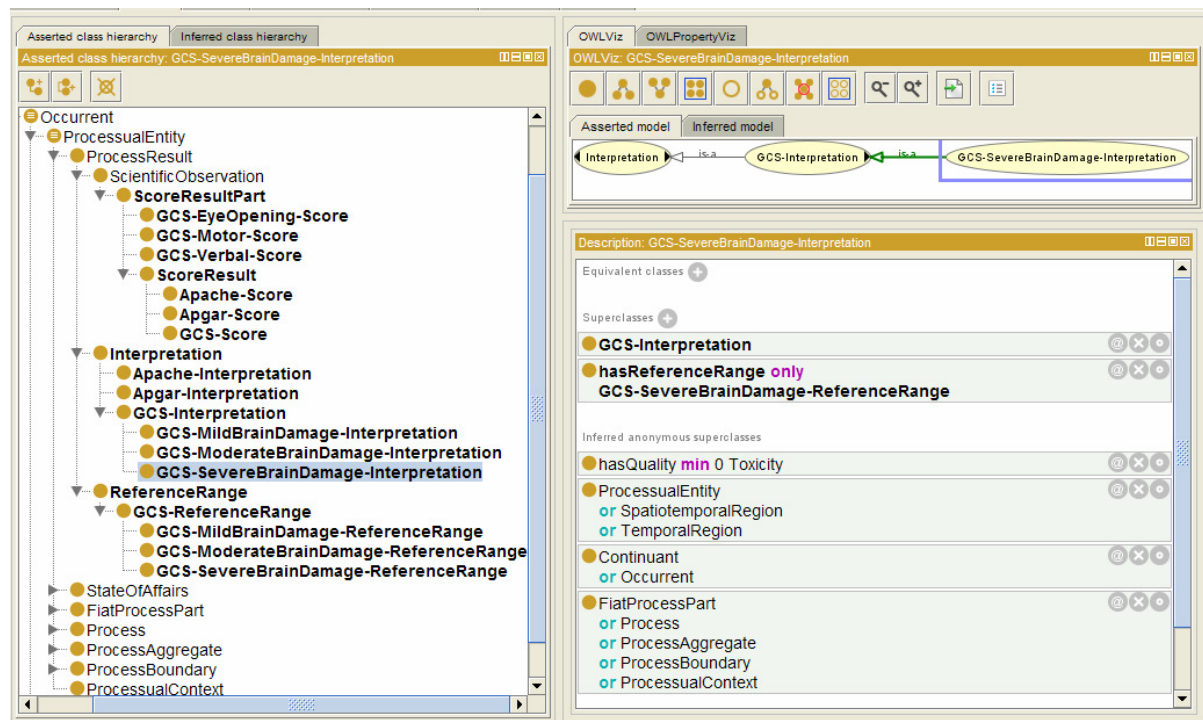


Abbildung 64: Reference Ranges für Scores auf Basis von ACGT

Die Relationen zwischen den Konzepten werden dann über ObjectProperties realisiert.

### 12.3.2. Integritätsregeln für das OWL-Modell

Die Instanziierung der Klassenhierarchie alleine ist für die Nutzung als Ontologie nicht ausreichend. OWL gestattet außerdem die Zuweisung von Einschränkungen an die Klassen, um die Relationen zu verifizieren. Das Beispiel in der nachfolgenden Abbildung

drückt aus, dass ein Score einen Befund haben muss. Zusätzlich erbt ein Score die notwendige Verknüpfung mit einem Code aus einem Katalogeintrag.

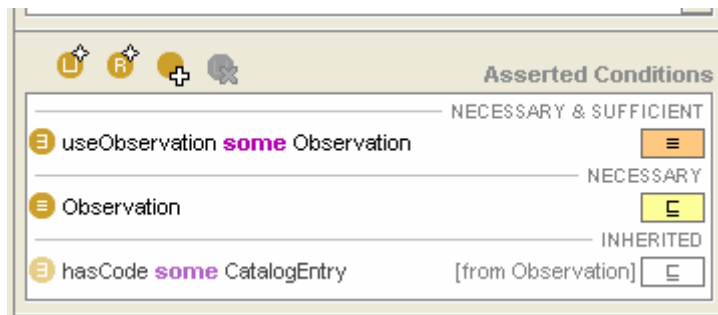


Abbildung 65: Einschränkungen für Scores in Protégé

Alle Relationen können mit Hilfe von Einschränkungen ausgedrückt werden, anhand derer Reasoner die Konsistenz der Ontologie überprüfen können.

### 12.3.3. Weitere Integration in ACGT

Zur Unterstützung des Mapping-Prozesses sollte zur Vereinfachung die bereits entwickelte Ontologie für Scores in ACGT integriert werden. Dazu sind die einzelnen Klassen entsprechend neu auszurichten. Im Falle des Konzeptes "Person" zeigt sich, dass das bereits vorhandene Konzept "HumanBeing" wiederverwendet werden kann, so dass dieses im Prinzip überflüssig ist. Somit ist zwischen einem eigenen Konzept "Person", die über "equivalentClass" mit "HumanBeing" unifiziert werden kann, und der direkten Wiederverwendung abzuwägen.

Die zur Identifikation notwendigen Codes und die damit verbundenen Codesysteme sind wie bei CSO als Informations-Objekt definiert und entsprechend aufgehängt. Hier bietet sich eine direkte Abbildung an. Die in Abbildung 63 aufgelisteten Codes und Codesysteme sind dem Glasgow Coma Scale (GCS) zugeordnet.

Die in der daran anschließenden Abbildung 64 differenzierten Details sind dem gleichen Score-System zugeordnet und repräsentieren die Interpretation, den Referenzbereich sowie die Details (Komponenten).

In OWL noch nicht adäquat repräsentiert sind die Werte für den Referenzbereich, da OWL (1.0) Einschränkungen<sup>42</sup> auf Datentypen nur für Instanzen und nicht für Klassen zulässt – zumindest nicht in OWL-DL. Daher muss dies durch eine explizite Modellierung über separate Klassen erfolgen.

Die Relationen zwischen den Konzepten nutzen dabei die gleiche Hierarchie von Relationen:

- hasRelationship
  - hasInterpretation

---

<sup>42</sup> Das hat sich mit OWL2 geändert, weil man eine Entität gleichzeitig als Klasse und Instanz betrachten kann. Die Semantik ist dann aber nicht mehr klar geregelt, so dass Reasoner hier Probleme haben und deshalb davon abzuraten ist.

- 
- hasReferenceRange
  - hasInvolvedPerson
    - hasAuthor
    - hasObserver
  - hasValue
    - hasCode
    - hasEntry
    - hasScoreValue
  - inverseOf hasValue
    - isValueFor
    - isValueOf
  - isPartOf

## 13. Mapping

Die Harmonisierung zwischen Ontologien, auch Ontology Coordination genannt, kann auf verschiedene Weise geschehen: Matching, Mapping, Alignment (Mediation), Merging. Diese verschiedenen Mechanismen adressieren unterschiedliche Ontologie-Komponenten, d.h., sie geschehen auf Klassenebene oder zumeist auf ObjectProperty-Ebene, wobei die Harmonisierung selbst stets auf der Instanzenebene entschieden wird.

Während beim Matching direkte Übereinstimmungen gesucht werden, erfolgt beim Mapping ein Ähnlichkeitsvergleich auf der Basis der zugrunde liegenden Konzepte. Beim Alignment erfolgt eine Angleichung an ein gemeinsames Konzept, während beim Merging die Konzepte (Concatination) verknüpft werden [BloOem2009].

In diesem Abschnitt der Arbeit werden die Grundlagen und Mechanismen des Mappings zwischen CSO und der Mediatorontologie ACGT bzw. der ausgearbeiteten Ergänzung ACGT-FO näher erläutert.

### 13.1. Basiskonzepte

Nachfolgend werden anhand eines Namens und einer Telefonnummer die Grundlagen des Mappings erklärt. Im Folgenden wird der Begriff Mapping generisch für die verschiedenen Koordinierungsmechanismen verwendet.

#### 13.1.1. Beispiel-Mapping anhand des Patientennamens

Abbildung 66 zeigt ein Beispiel eines direkten Mappings zwischen HL7 Version 2.x PID-5 und einem Namen in der Version 3:

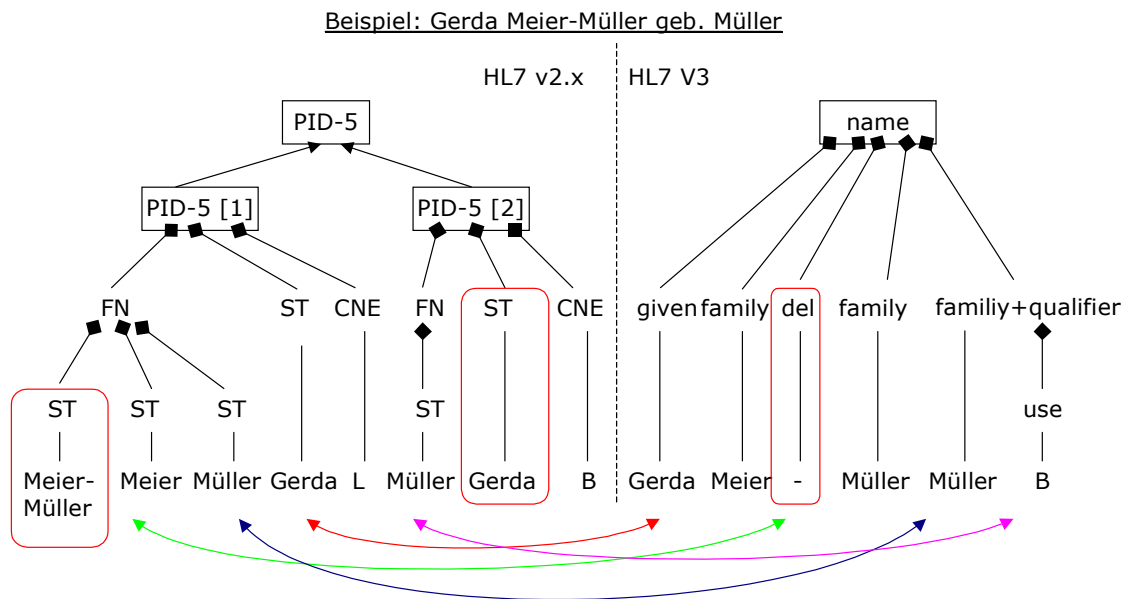


Abbildung 66: Beispiel-Mapping V3-v2.x (als direktes Mapping)

Wie durch die rot markierten Stellen ersichtlich ist, werden nicht alle Informationen in beiden Modellen explizit dargestellt. So wird der Delimiter in Version 3 beispielsweise in der zusammengesetzten Darstellung des Familiennamens "versteckt".

Für Basiskonzepte muss ein hierarchisches Mapping realisiert werden, d.h. eine derartige Umsetzung ist nur einmal zu definieren und dann auf andere Vorkommen entsprechend zu übertragen. Im Prinzip sollte ein "verschachteltes Mapping" an jeder Stelle möglich sein.

Nachfolgend nun die gleiche Umsetzung über ACGT als Mediator:

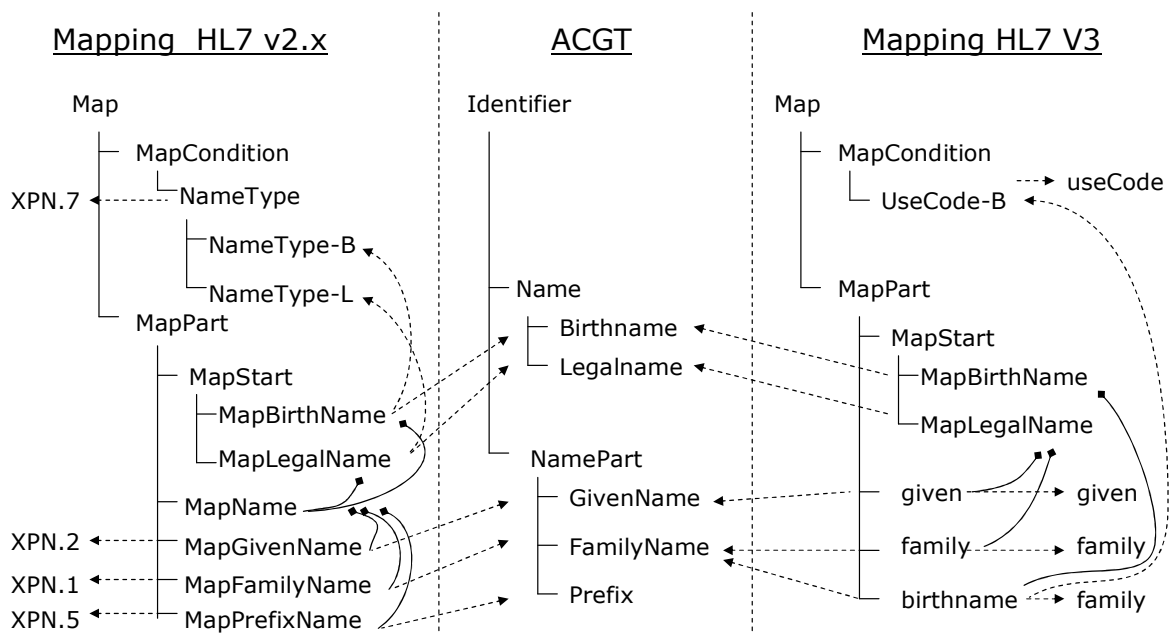


Abbildung 67: Beispiel-Mapping V2.x-ACGT-V3 (indirektes Mapping)



### 13.1.2. Beispiel-Mapping anhand der Telefonnummer

Als ein Beispiel für das hierarchische Mapping soll eine Telefonnummer dienen. Daran kann die Umsetzung demonstriert werden. Hierbei kommen zwei Problemfelder zum Tragen. Zum Einen haben die beiden HL7-Standard-Versionen eine unterschiedlich Granularität, was die Informationseinheiten betrifft. Zum Anderen haben dieselben Bedeutungen (Konzepte) eine unterschiedliche Repräsentation.

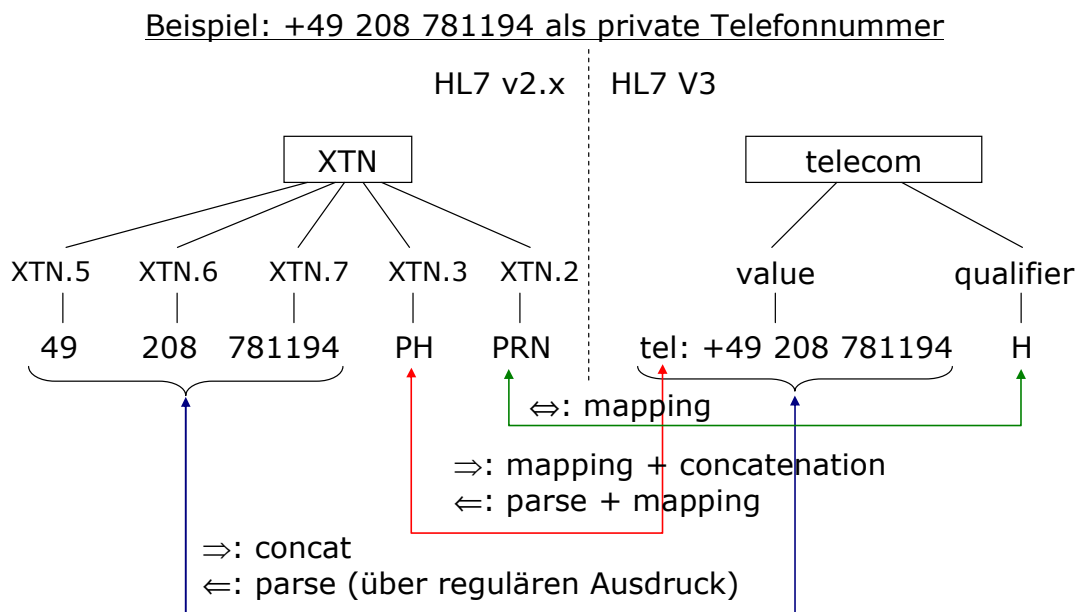


Abbildung 68: Beispiel-Mapping V3-v2.x (mit Zerlegung)

Eine detailliertere Betrachtung erfolgt im Zusammenhang mit regulären Ausdrücken, die notwendig sind, um „prä-koordinierte“ Konzepte in einzelne Bestandteile zu zerlegen.

## 13.2. Darstellung in Beschreibungslogik

Wenn wir unter Identität einer Person die Kombination der Merkmale verstehen, anhand derer sich ein Individuum von anderen unterscheiden lässt, dann sind gegebene Merkmale mitunter nicht eindeutig (z.B. Name). Auch natürliche Merkmale wie das Geburtsdatum reichen für eine Identifizierung oft nicht aus, da sie z.B. zu grob sind. Abhilfe schaffen verwaltete eindeutige Identifikatoren (z.B. in der verwalteten Domäne eindeutige ID-Nummern), aber auch einzigartige Eigenschaften oder Biometrien. Oft wird eine Kombination aus Merkmalen verwendet, was die Diskriminanz verbessert. Im Folgenden werden beispielhaft die in HL7 standardisierten Merkmale und Parameter betrachtet.

Das nachfolgende Beispiel besagt, dass ein Name aus Vorname und Nachname besteht und einen bestimmten Typ hat. Ein legaler Name oder ein Display-Name hingegen schränken den NameType auf einen bestimmten Wert ein.

$\forall n: n \in \text{Name} \rightarrow$

$(\exists a: a \in \text{Nachname} \wedge \text{hasPart}(n,a)) \wedge$

$(\exists b: b \in \text{Vorname} \wedge \text{hasPart}(n,b)) \wedge$

$(\exists c: c \in \text{NameType} \wedge \text{hasPart}(n,c))$

mit

$\forall l: l \in \text{LegalName} \rightarrow$

$(l \in \text{Name} \wedge \text{hasType}(l,t) \wedge \text{isType}(t, "L"))$

und

$\forall d: d \in \text{DisplayName} \rightarrow$

$(d \in \text{Name} \wedge \text{hasType}(d,t) \wedge \text{isType}(d, "D"))$

Ein Mapping von Konzepten verschiedener Ontologien kann wie folgt realisiert werden (Abbildung 69):

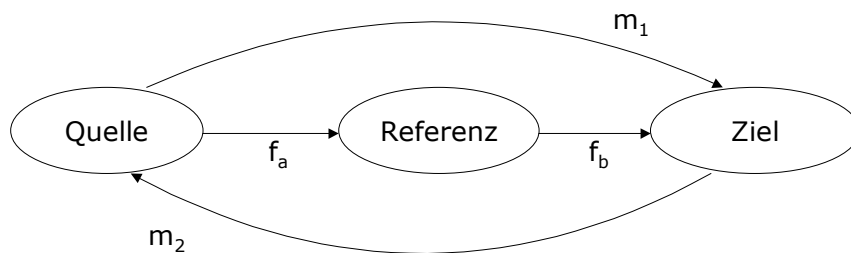


Abbildung 69: Mapping

Für äquivalentes Mapping gilt die Bijektivität atomarer Konzepte:

$$f_a^{-1}(f_a(i)) = i$$

bzw.

$$f_b^{-1}(f_b(i)) = i$$

Über die Transitivität des Mappings resultiert folgende Gleichung:

$$m_1 = f_b(f_a(i))$$

$$m_2 = f_a^{-1}(f_b^{-1}(i))$$

Ein nicht äquivalentes Mapping muß über Bedingungen realisiert werden. Dazu kommen dann die bereits eingeführten Funktionen für „Split“ und „Join“, um in den Kommunikationsstandards „nicht-atomare Konzepte“ weiter handhaben zu können.

### 13.3. Ablauf des Mappings

Das vorangehend erläuterte Mapping der Konzepte in die Fachdomäne ist in Abbildung 70 durch die gestrichelten Pfeile symbolisiert. Durch die Notwendigkeit, Bedingungen und Abhängigkeiten zu berücksichtigen, sind Mappings ebenfalls als separate Konzepte realisiert worden.

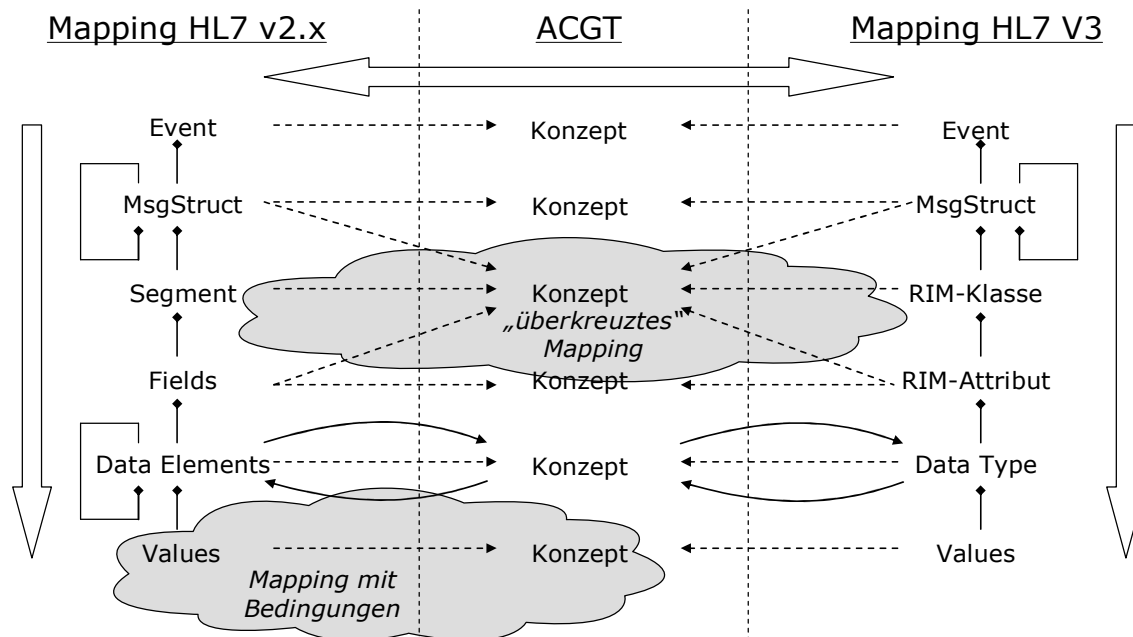


Abbildung 70: Komponentenorientiertes Mapping

Die Vorgehensweise ist im Prinzip Top-Down, d.h. die Konzeptzusammenhänge werden von oben nach unten abgearbeitet. Hierbei spielen die hierarchischen Relationen, die primär auf der part-of-Relation basieren, die Hauptrolle, denn unter Berücksichtigung der Qualitäten sowie der hinterlegten Bedingungen müssen diese durch einen entsprechenden Agenten ausgewertet werden.

Während dieses Vorgangs kommt die Umkehrrelation zum Tragen, da je nach Zusammenhang eine Relation nur von einer Seite definiert wurde. (Siehe hierzu beispielsweise die Problematik der Datentypen, auf die sowohl die Datenelemente als auch die Komponenten verweisen (Abbildung 49). So muss beim Mapping die Komponentenbeziehung ausgewertet werden, die aber nur von der Seite der Komponente definiert worden ist.)

Gestartet wird auf der obersten Ebene. Dabei werden die Events und die Nachrichtenstrukturen aufeinander abgebildet. Von da aus wird beidseitig vorgegangen, da inhaltsäquivalente Strukturen aufgebaut werden müssen.

### 13.4. Ergänzung zu CSO

Die notwendigen Konzepte für das Mapping sind in CSO nicht enthalten und werden deshalb hier separat als eigene Ontologie aufbauend auf CSO definiert.

Tabelle 17: Addendum zu CSO

Element (Klasse)	Erläuterung
Thing	
entity (bfo) continuant (bfo) dependent_continuant (bfo) generically_dependent_continuant (bfo)	
InformationObject (cso)	Für dieses Konzept könnte auch das Konzept aus ACGT genutzt werden
Mapping (cso-fo) MapCondition (cso-fo) MapPart (cso-fo) MapStart (cso-fo)	
RegExpressionPart (cso-fo) RegExpressionParameter (cso-fo) RegExpression (cso-fo)	Zerteilen eines nicht-atomaren Elementes mit Hilfe regulärer Ausdrücke
specifically_dependent_continuant (bfo)	
independent_continuant (bfo) occurent (bfo)	

Die Konzepte werden nachfolgend eingehender erläutert.

### 13.4.1. Mapping

Wie in "Abbildung 31: Ontologie-Struktur" dargestellt kann ein Mapping auf dieser Ebene zwischen den verschiedenen Kommunikationsstandards vermitteln. Weitere Details dazu werden nachfolgend erörtert.

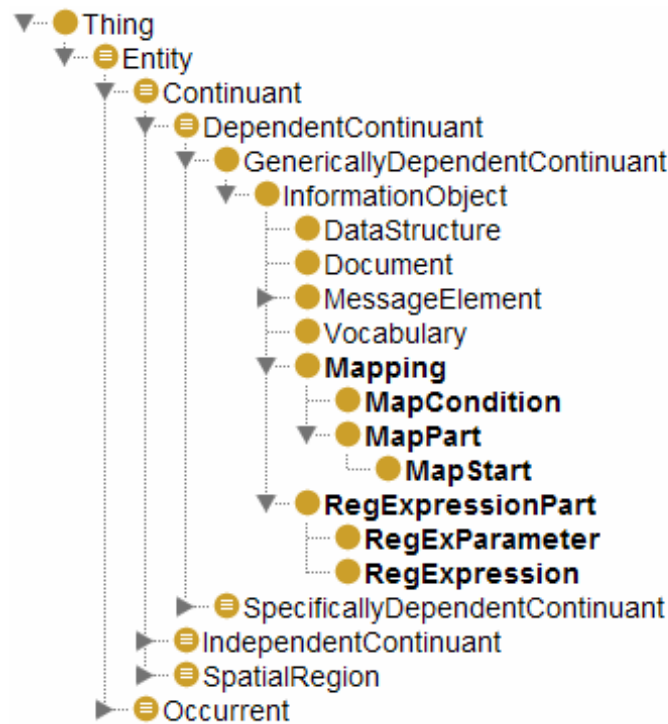


Abbildung 71: Basis-Ontologie: Mapping (Darstellung in Protégé)

Dazu gehören auch die entsprechenden Relationen:

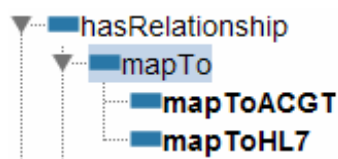


Abbildung 72: Basis-Ontologie: Relationen zum Mapping (Darstellung in Protégé)

Da eine Relation als Mapping wegen der zugrunde liegenden Komplexität (Bedingungen) als eigenständiges Konzept ausgedrückt werden muss, sind zwei separate Object-Properties zu beschreiben: `mapToACGT` zeigt auf das zugehörige Konzept in ACGT, während `mapToHL7` auf das entsprechende Konzept in der HL7-Ontologie verweist.

### 13.4.2. RegExpression

Wie bereits angedeutet reicht ein direktes Mapping nicht aus, da nicht sichergestellt werden kann, dass die Granularität der Konzepte in den jeweiligen Kommunikationsstandards übereinstimmt. Daher muss ein Mechanismus etabliert werden, der eine weitere Aufteilung ermöglicht. Dies wird über „*Regular Expressions*“ erreicht:

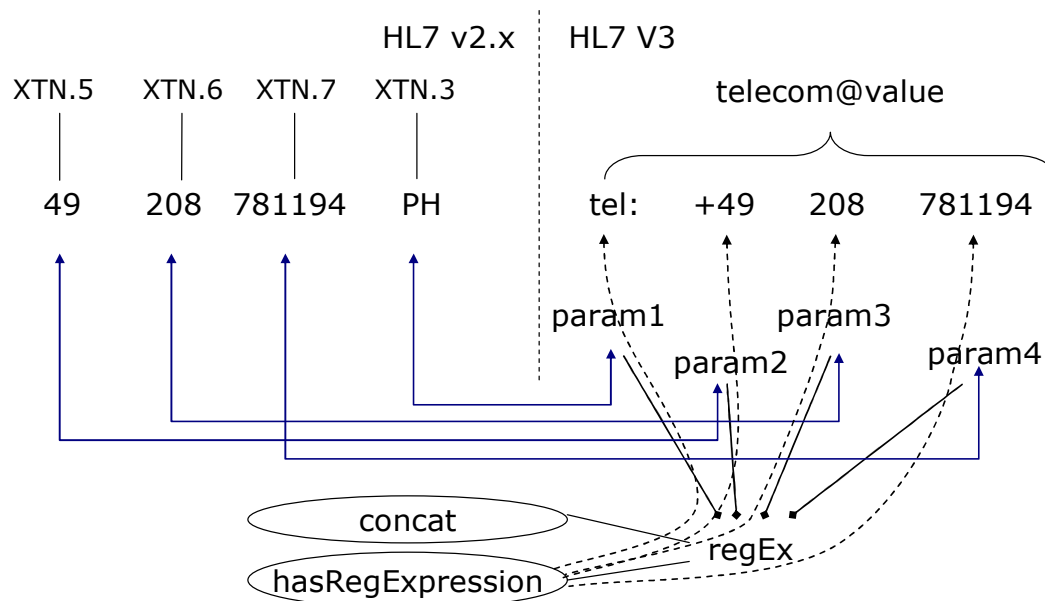


Abbildung 73: Funktionsweise der Regular Expressions

Eine solcher Ausdruck („RegEx“) enthält entsprechende Komponenten („paramX“), die jeweils über das DatatypeProperty „paramNo“ eine entsprechende Zuordnung in der Reihenfolge bekommen, die durch das Attribut „hasRegularExpression“ (vgl. Abbildung 77) ausformuliert wird. Für das obige Beispiel würde der Ausdruck wie folgt lauten:

`(tel|mailto):([[+].[/-]].[/-]).)`

Umgekehrt sorgt das Attribut „concat“ dafür, die Bestandteile (Komponenten/Parts) wieder zusammzusetzen, da ein regulärer Ausdruck dafür nicht verwendet werden kann. Die Umsetzung über eine separate Angabe regelt zusätzlich auch die exakte Darstellung, während ein regulärer Ausdruck Varianten in der Darstellung berücksichtigt. Die aktuelle Realisierung sieht nur einen regulären Ausdruck vor. Es kann aber notwendig werden, hier mehrere Ausdrücke (als Liste) zu hinterlegen.

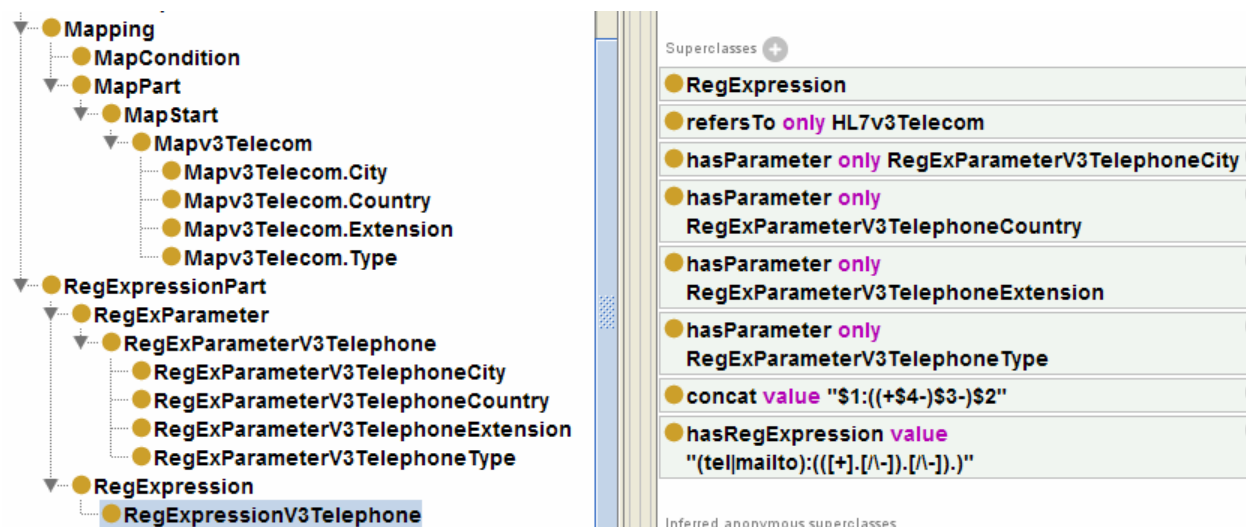


Abbildung 74: Regular Expressions (Darstellung in Protégé)

Das in vorhergehender Abbildung markierte Konzept ist nachfolgend in der OWL-Darstellung angegeben:

```

<owl:Class rdf:about='&hl7-v3;RegularExpressionV3Telephone'>
  <rdfs:comment xml:lang="en">specifies the regex for telephone numbers</rdfs:comment>
  <rdfs:subClassOf>
    <owl:Class rdf:about='&map;RegularExpression' />
  </rdfs:subClassOf>

  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasRegularExpression" />
      <owl:hasValue rdf:datatype="&xsd:string">
        (tel|mailto):(([\+].[/\-\]).[/\-\]).)
      </owl:hasValue>
      <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#concat" />
      <owl:hasValue rdf:datatype="&xsd:string">$1:(+$4-$3-$2</owl:hasValue>
      <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>

  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:FunctionalProperty rdf:about="&cso;refersTo"/>
      </owl:onProperty>
      <owl:allValuesFrom>
        <owl:Class rdf:about="&hl7-v3;HL7v3Telecom"/>
      </owl:allValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>

  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:FunctionalProperty rdf:about="&map;hasParameter"/>
      </owl:onProperty>
      <owl:allValuesFrom>
        <owl:Class rdf:about="&hl7-v3;RegExParameterV3TelephoneType"/>
      </owl:allValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>

  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:FunctionalProperty rdf:about="&map;hasParameter"/>
      </owl:onProperty>
      <owl:allValuesFrom>
        <owl:Class rdf:about="&hl7-v3;RegExParameterV3TelephoneCountry"/>
      </owl:allValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>

  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:FunctionalProperty rdf:about="&map;hasParameter"/>
      </owl:onProperty>
      <owl:allValuesFrom>
        <owl:Class rdf:about="&hl7-v3;RegExParameterV3TelephoneCity"/>
      </owl:allValuesFrom>
    </owl:Restriction>
  </rdfs:subClassOf>

```

# Entwicklung einer ontologiebasierten Architektur zur Sicherung semantischer Interoperabilität zwischen Kommunikationsstandards im Gesundheitswesen

```
<rdfs:subClassOf>  
  <owl:Restriction>  
    <owl:onProperty>  
      <owl:FunctionalProperty rdf:about="&map;hasParameter"/>  
    </owl:onProperty>  
    <owl:allValuesFrom>  
      <owl:Class rdf:about="&hl7-v3;RegExParameterV3TelephoneExtension"/>  
    </owl:allValuesFrom>  
  </owl:Restriction>  
</rdfs:subClassOf>  
</owl:Class>
```

Die Parameter für Regular Expressions können ebenfalls in den Mapping-Prozess mit einbezogen werden, wie Abbildung 75 demonstriert:

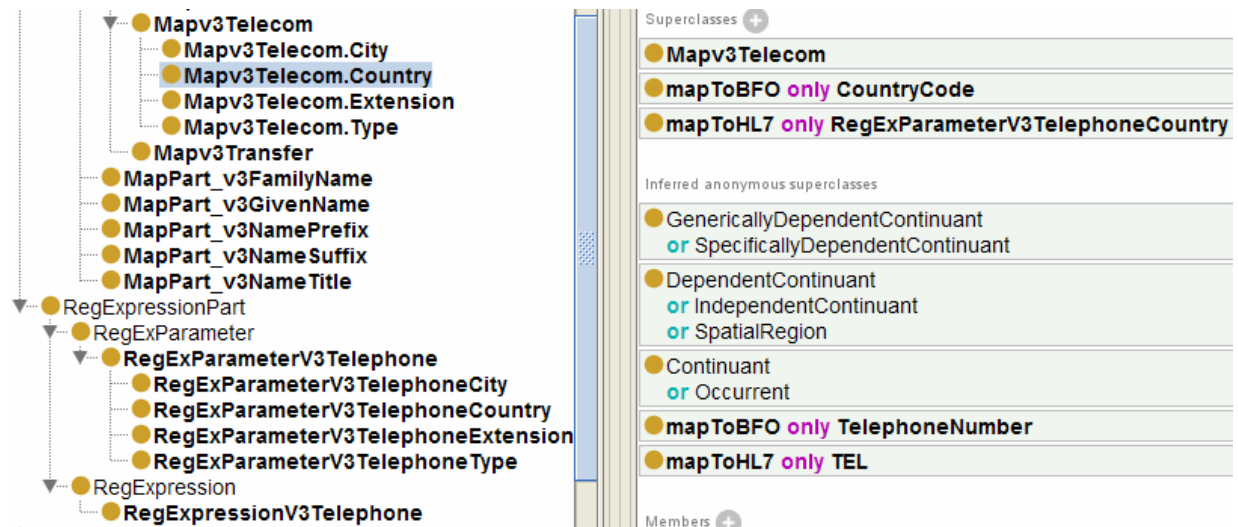


Abbildung 75: Mapping auf Regular Expressions (Darstellung in Protégé)

Die Relationen lassen sich auch graphisch darstellen:

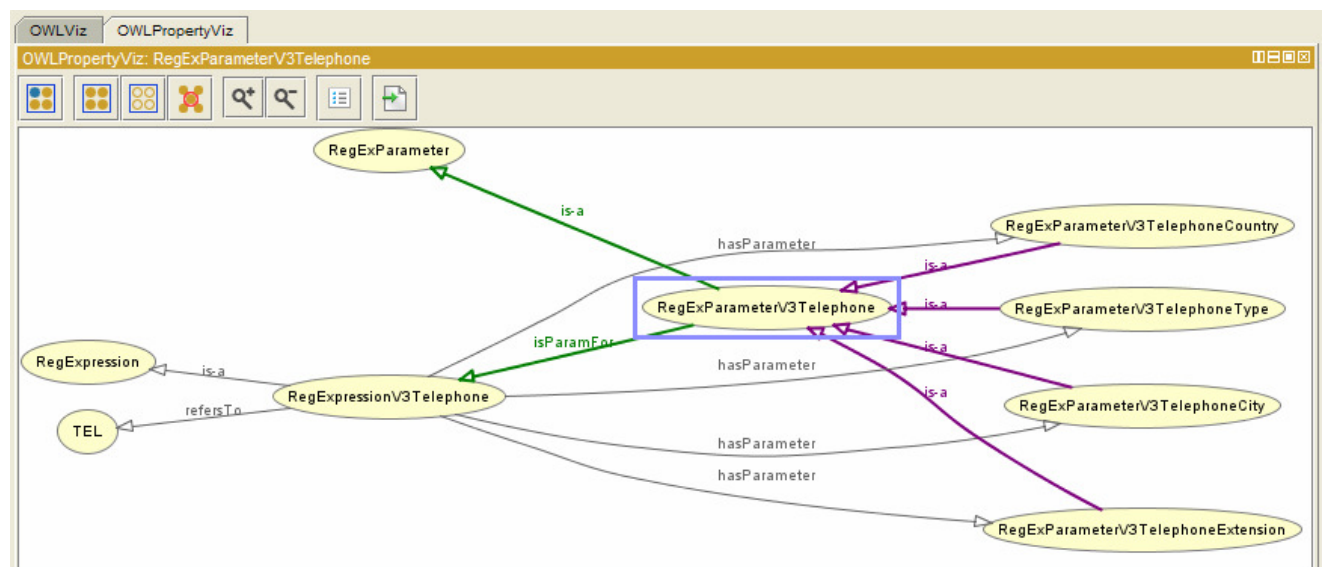


Abbildung 76: graphische Darstellung der Regular Expressions (Darstellung in Protégé)



### 13.4.3. Data-Properties

Die meisten Informationen werden als eigenständige Klassen modelliert, so dass die Relationen dazwischen über die ObjectProperties dargestellt werden können. Übrig bleibt das conditionStatement, das die eigentliche Bedingung innerhalb des Mapping-Prozesses enthält:

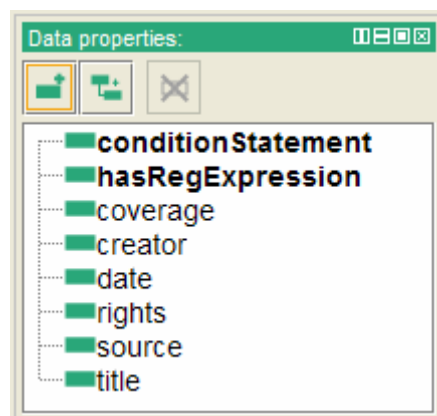


Abbildung 77: Data-Properties (Darstellung in Protégé)

Das hier aufgelistete Property "hasRegularExpression" wird benötigt, um einen regulären Ausdruck (als Zeichenkette) für den Mapping-Prozess zu hinterlegen.

### 13.4.4. Mapping für v2.x

Nachfolgend eine einfache Übersicht über das Mapping von HL7 v2.6 auf ACGT:

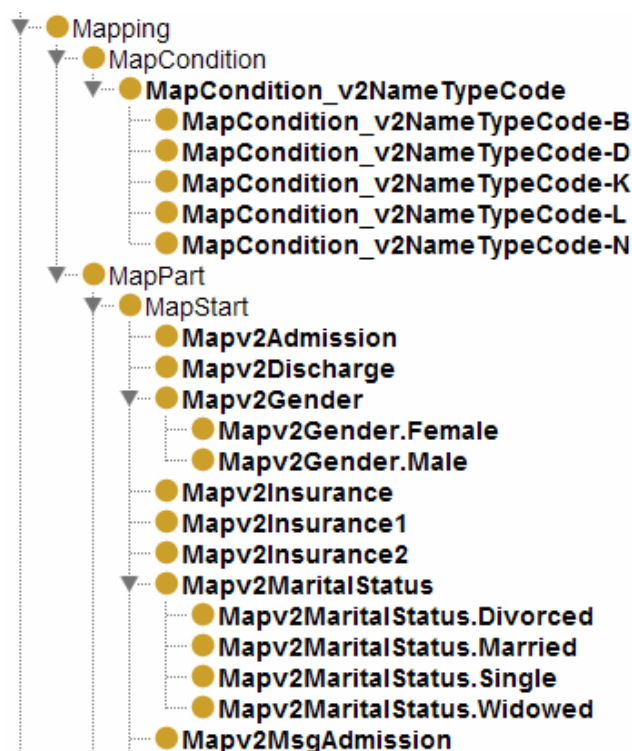


Abbildung 78: v2-Mapping (Darstellung in Protégé)

Ein einfaches Mapping lässt sich wie folgt am Beispiel "Female (ACGT)" auf "F (v2.x)" darstellen. Hierbei werden die ObjectProperties MapToACGT und MapToHL7 entsprechend befüllt:

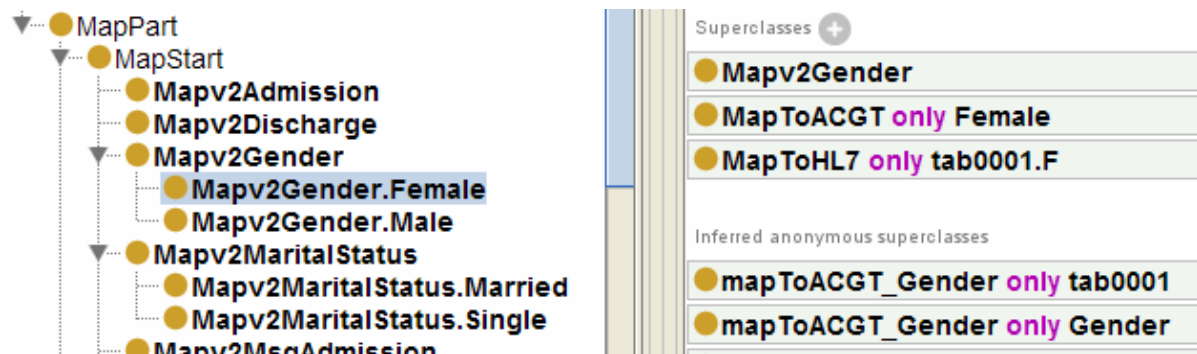


Abbildung 79: v2-Mapping (Darstellung in Protégé)

Das Mapping eines Namens mit der Auswertung von Bedingungen ist schon etwas komplizierter:

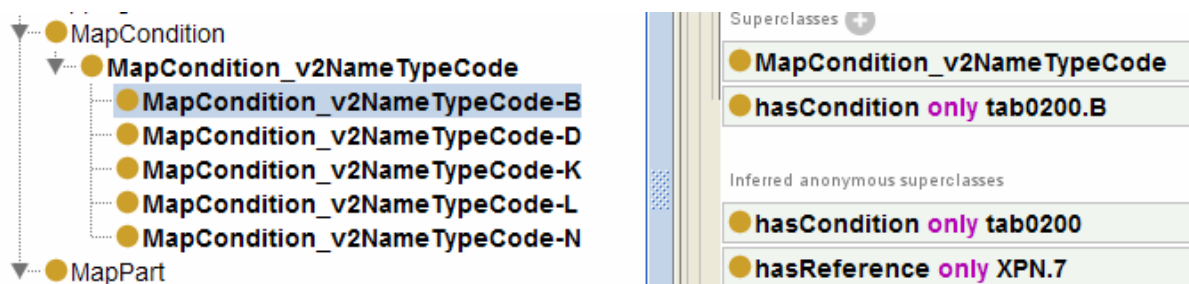


Abbildung 80: v2-Mapping mit Bedingungen (Darstellung in Protégé)

Die Beziehungen lassen sich aber auch graphisch darstellen:

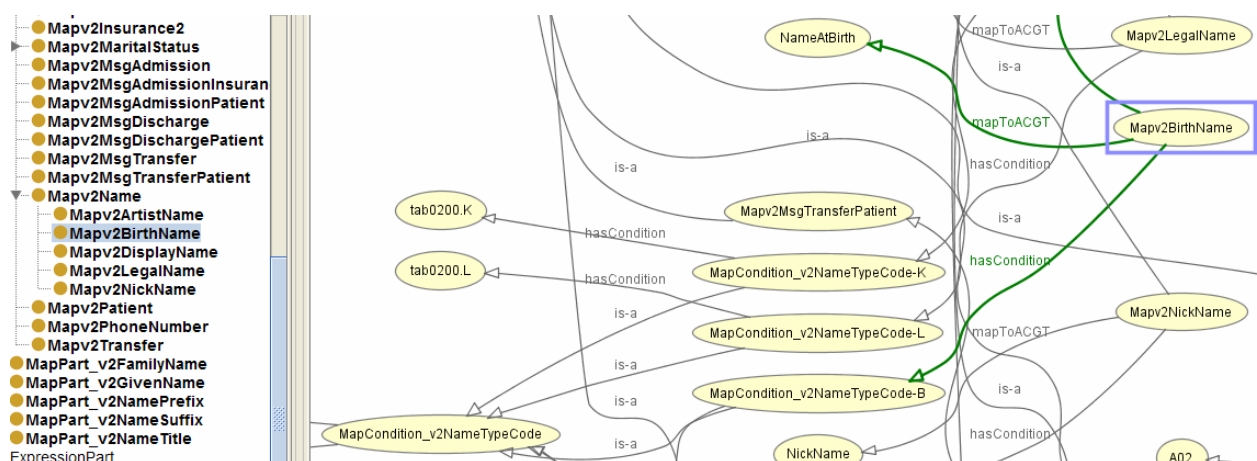


Abbildung 81: v2-Mapping mit Bedingungen (Darstellung mit OwlPropViz)

### 13.4.5. Mapping für V3

Der entsprechende Mappingprozess in HL7 V3 sieht ähnlich aus:

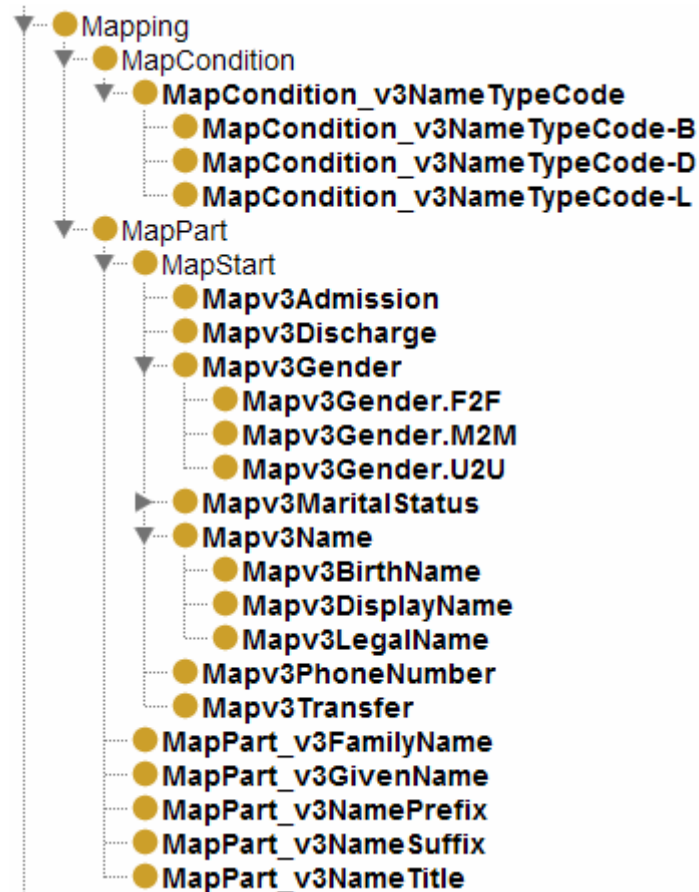


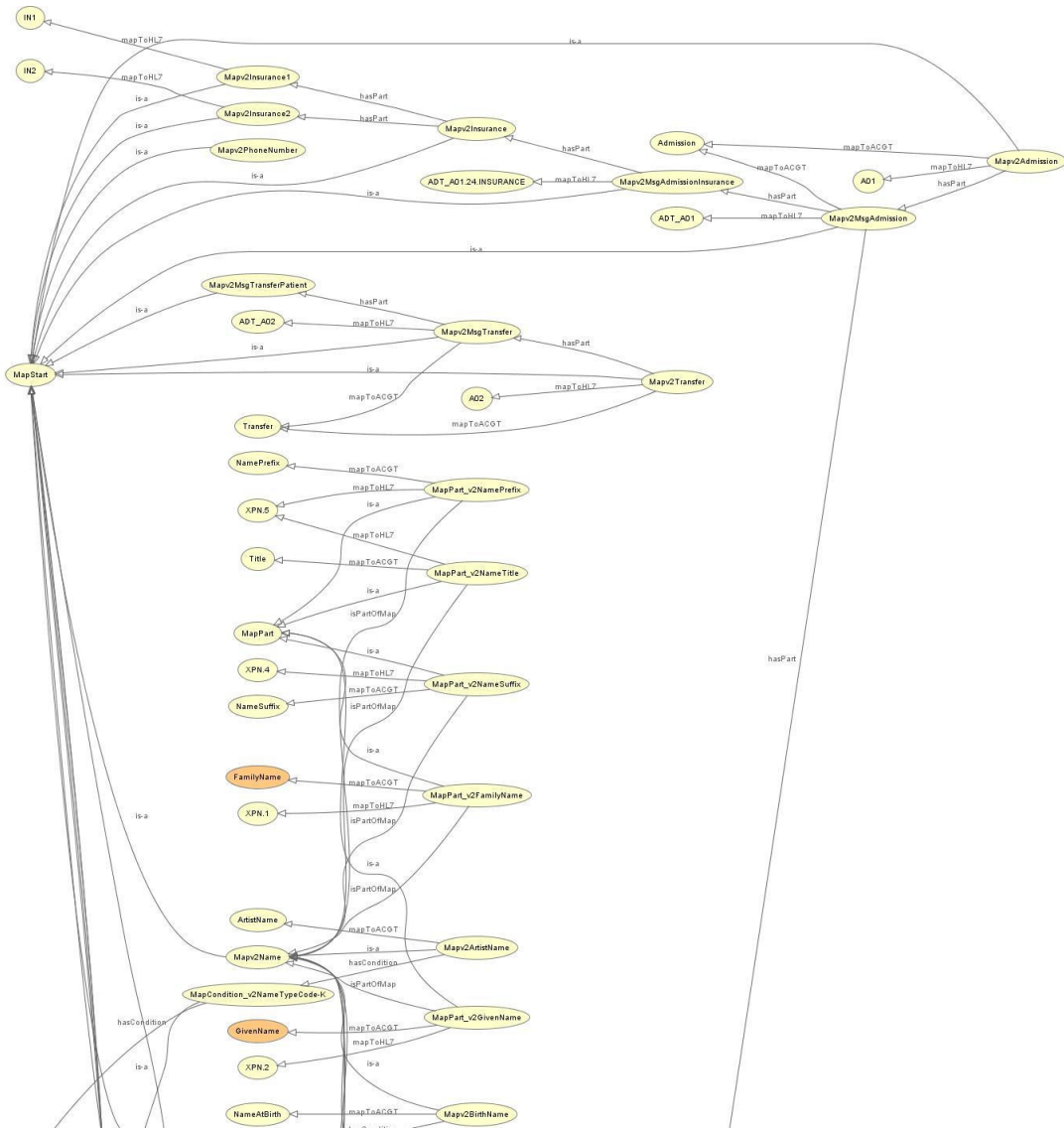
Abbildung 82: Übersichtsschaubild (Darstellung in Protégé mit OwlPropViz)

Auch hier werden die einzelnen Klassen/Konzepte durch entsprechende ObjectProperties zueinander in Beziehung gesetzt.

### 13.4.6. Übersichtsschaubild

Eine Auswahl der Beziehungen lassen sich mit Hilfe des Plug-Ins OWLPropViz [OWLPropViz, Wach2008] wie folgt visualisieren:

# Entwicklung einer ontologiebasierten Architektur zur Sicherung semantischer Interoperabilität zwischen Kommunikationsstandards im Gesundheitswesen



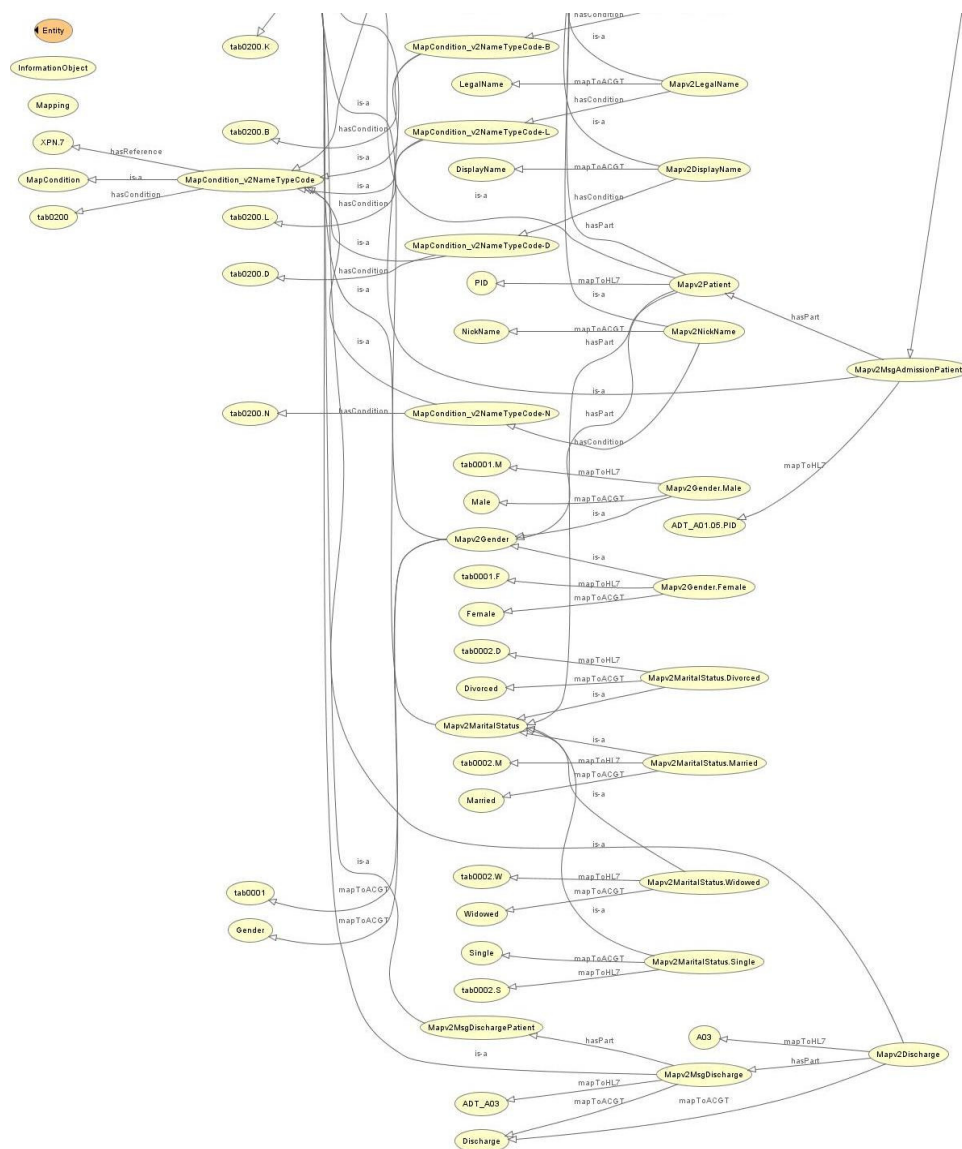


Abbildung 83: V3-Mapping (Darstellung in Protégé)

Für diese Übersicht sind insbesondere die für das Mapping relevanten Relationen zur Visualisierung ausgewählt worden.

## 14. praktische Erzeugung der OWL-Dateien

Dieser Abschnitt der Arbeit beschäftigt sich mit den unterschiedlichsten Aspekten, die bei der praktischen Umsetzung auftreten. Zum Einen werden die verschiedenen Möglichkeiten eruiert, um aus den vorhandenen Quelldateien OWL-Dateien zu erzeugen. Zum Anderen werden aber auch die Schwierigkeiten beim Mapping sowie die Randbedingungen durch den Einsatz des gewählten Tools (Protégé) beschrieben.

### 14.1. Erzeugung der Ontologien

Wie aus den Erläuterungen in den vorhergehenden Abschnitten zu entnehmen ist, muss die finale Ontologie "umfassend" sein, d.h. die anderen Ontologien inkludieren. Zu

diesem Zweck werden die beiden "Master-Mapping-Ontologien" für v2.x und V3 definiert, die über Imports als einzigem Mechanismus zur Zusammenführung von Ontologien in OWL die anderen Ontologien zusammenführen, die wiederum den Importmechanismus benutzen.

Die Basis-Ontologien für HL7 v2.x und V3 werden mit Hilfe von Programmen erzeugt. Details zu diesem Prozess (Lösungsansätze, Vor- und Nachteile, etc.) sind nachfolgend detaillierter beschrieben.

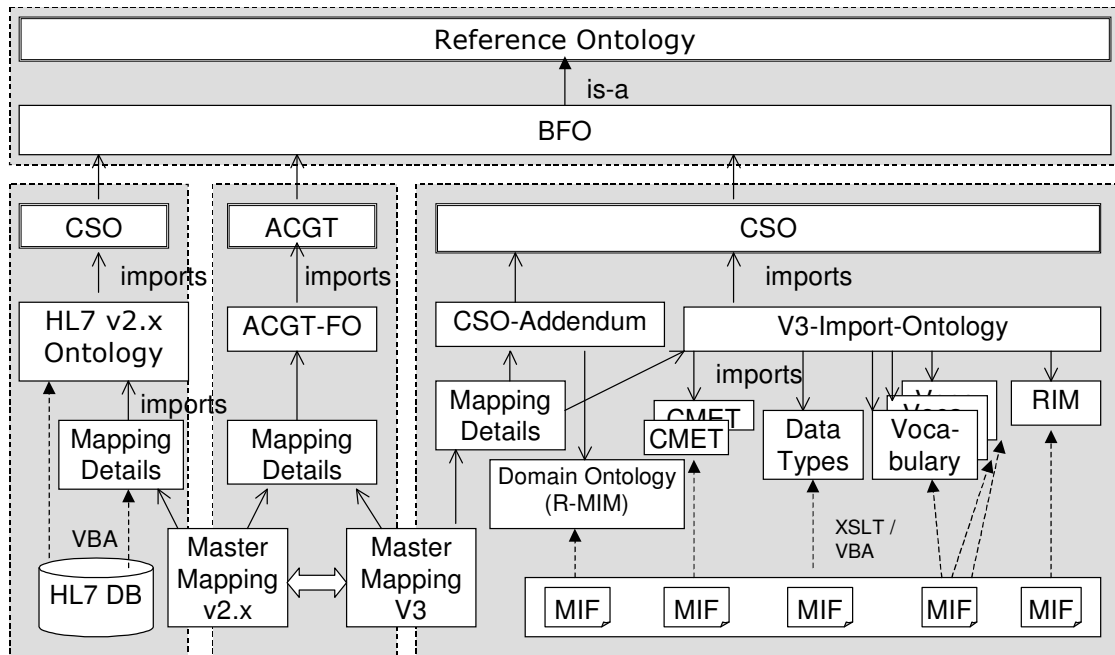


Abbildung 84: Erzeugung der Ontologien

Die Import-Struktur lässt sich wie folgt daraus extrahieren, wobei die Pfeile auf die notwendigen Imports verweisen:

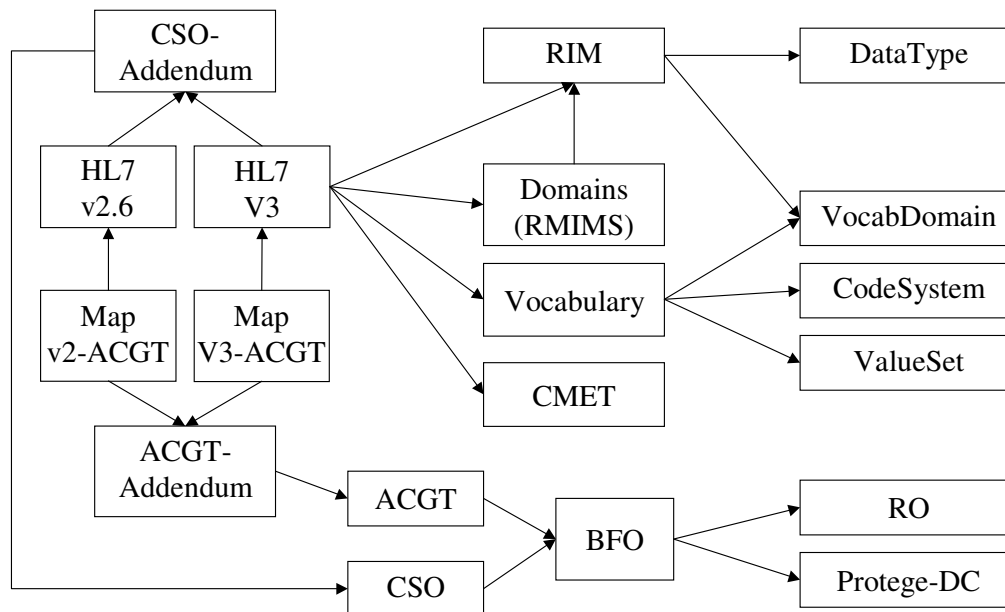


Abbildung 85: Ontologie-Import-Struktur

### 14.1.1. Ontologiegundstruktur für die Kommunikationsstandards

Die Ontologiegundstruktur für die Kommunikationsstandards berücksichtigt folgende Aspekte:

Tabelle 18: Ontologiegundstruktur für die Kommunikationsstandards

Struktur	Top-Level-Ontologie	HL7 v2.x	HL7 V3
InformationObjectQualities	X		
CodingStrength	X	X	X
Condition	X		
NullFlavor	X		X
DeleteIndication	X	X	
Optionality	X	X	X
Table Type	X	X	
Truncation	X	X	
Dynamic Aspects (role + process)	X		
Actor	X		
Event	X	X	
Interaction			
Static Aspects (Inform.Obj.)	X		
Data Structure		X	
Message Element	X		
Component		X	
Data Element		X	

Struktur	Top-Level-Ontologie	HL7 v2.x	HL7 V3
Field		X	
Message Structure		X	
Message		X	
Message Structure Element		X	
Segment		X	
CMET-Part			X
CMET			X
RIM-Part			X
RIM-Class			X
RIM-Attribute			X
RIM-Association			X
R-MIM-Part			X
R-MIM-Class			X
R-MIM-Attribute			X
CMET			X
Vocabulary	X		
VocabularyTable		X	
TableValue		X	
codeSystem			X
conceptDomain			X
valueSet			X

### 14.1.2. Ontologiegrundstruktur für das Mapping

Hier sind folgende Aspekte relevant:

Tabelle 19: Ontologiegrundstruktur für das Mapping

Struktur	Top-Level-Ontologie (Addendum)	HL7 v2.x	HL7 V3
Mapping	X		
MapCondition	X	X	X
MapPart	X	X	X
MapStart	X	X	X
RegExpressionPart	X		
RegExparameter	X		X
RegExpression	X		X



## 14.2. Repräsentation von Ontologien mittels OWL

Die Umsetzung der in Abbildung 84 beschriebenen Struktur hat einige Probleme aufgeworfen, die im nachfolgenden besprochen und bewertet werden sollen.

### 14.2.1. Scripting

Die zu verwendenden Ontologien können nicht manuell, sondern müssen automatisch mit Hilfe eines Generators erzeugt werden. Daher stellt sich die grundlegende Frage, in welcher Form die Daten über die HL7-Standards vorliegen und wie daraus Ontologien generiert werden können?

Die besagten Daten liegen in Form einer relationalen Datenbank (MS Access für v2.x, [OemDud1996]) sowie als Satz von XML-Dateien (MIF-Files [HL7-MIF] bzw. als Einträge in der "Publication Database for V3" [HL7-PubDb] für V3) vor. Damit bieten sich konventionelle Programmiersprachen (VB-Script, VBA, Java, etc.) zur Konvertierung der Daten an. Bei den XML-basierten Daten stand außerdem die Überlegung im Raum, eine Konvertierung mit Hilfe von XSLT vorzunehmen. In der folgenden Tabelle werden die verschiedenen Generierungsmechanismen verglichen und bewertet.

Tabelle 20: Vergleich und Bewertung der Umsetzungsansätze

Std.	Grund- Lage	Format	Bewertung der Konvertierung mit			
			VBA	Java	VB-Script	XSLT
	All- gemein		Es hat sich gezeigt, dass zusätzliche Steuerungs- informationen (beispiels- weise Selektionen zur Filterung) zur Ausgabe in die OWL-Datei herange- zogen werden müssen. Darüber hinaus gibt es verschiedene Möglich- keiten zur Umsetzung als Ontologie (Klasse, In- stanz), welche durch eine graphische Ober- fläche leichter gesteuert werden können.		Prinzipiell gleich- wertig zu VBA, je- doch ist dort die Nutzung von GUI- Elementen sowie das Debuggen einfacher.	Die notwendige Para- metrierungsmöglich- keiten erhöhen die Pro- grammkomplexität sehr stark. Die Programmierlogik von XSLT divergiert zu den anderen Sprachen, so dass ein relativ hoher Aufwand bei der zukünftigen Wartung entsteht. Eine Selektion von De- tails ist nur sehr schwer zu realisieren. Größere Statements/ Programmkonstrukte (if-then-else, select, while) sind nur um- ständlich realisierbar.

Std.	Grund-Lage	Format	Bewertung der Konvertierung mit			
			VBA	Java	VB-Script	XSLT
				vergleichbar mit VBA; jedoch ist die IDE von Microsoft für VBA leichter zu debuggen		
v2.x	HL7-DB	rel. DB (MS Access)	GUI mit Programmlogik kombinierbar und debugbar. Erfahrung für andere Generatoren (HTML, XSD) liegt bereits vor.	Integration fehlt.		nicht anwendbar
	HL7 v2.xml	XML-Schema	Die Informationen sind über mehrere Dateien verteilt, daher schwierig zu verarbeiten. Daten liegen nicht mehr im originären Format vor.			
V3	Pub-DB	rel. DB (MS Access)	wird zukünftig durch die MIF-Files abgelöst, daher ist diesen der Vorzug zu geben.			
						nicht anwendbar
	MIF	XML-Dateien	Diese existieren in unterschiedlichen Versionen. In der Umsetzung wurde die neueste (MIF2) verwendet.			
						Verteilung der Informationen über mehrere Dateien ist schwierig zu handhaben.
						Vorteilhaft ist die Nutzung von XPath-Ausdrücken zum Zugriff

Std.	Grund- Lage	Format	Bewertung der Konvertierung mit			
			VBA	Java	VB-Script	XSLT
						auf referenzierte Elemente, was in den anderen Programmiersprachen deutlichen Aufwand verursacht.

Die Konvertierung der MIF-Files nach OWL wurde zuerst auf der Basis von XSLT realisiert, da dies als Methode der Wahl bei XML-Dateien angepriesen wird. Es wurde aber sehr schnell deutlich, dass ein relativ aufwändiges Programmgerüst entsteht, das nicht mehr ohne weiteres wartbar ist. Daher wurde auf der Basis der gemachten Erfahrungen ein neues Programm mittels VBA realisiert.

Die in der vorangehenden Tabelle aufgelisteten Kriterien haben zu einer Umsetzung mit Hilfe von VBA-Generatoren geführt. Die notwendigen Parameter wurden dabei entweder direkt in der Datenbank (bei HL7 v2.x) bzw. in Konfigurationsdateien (bei MIF-Files) hinterlegt. Entsprechende Oberflächen erlauben gleichzeitig eine leichte Kontrolle der Umsetzung (Abbildung 86 und Abbildung 87).

Hinsichtlich des Portierungsaspektes auf andere Betriebssysteme wäre aber Java der Vorzug zu geben. Die Nutzung der generierten OWL-Dateien auf anderen Plattformen ist hingegen gegeben.

### 14.2.2. HL7 v2.x

Der vorangegangenen Diskussion folgend, wird die Ontologie für HL7 Version 2.x durch ein entsprechendes VBA-Programm erzeugt. Die Details der Umsetzung werden hierbei über ein Formular gesteuert. Eine Umsetzung aller Details wäre für eine manuelle Verifikation zu umfangreich, so dass ein Ausschnitt festgelegt werden muss. Die Reduktion des Datenvolumens wird deshalb durch eine explizite Selektion geeigneter repräsentativer Details aus den entsprechenden DB-Tabellen manuell vorgenommen, um nicht zu hohe Lade- und Programmlaufzeiten zu erzeugen.

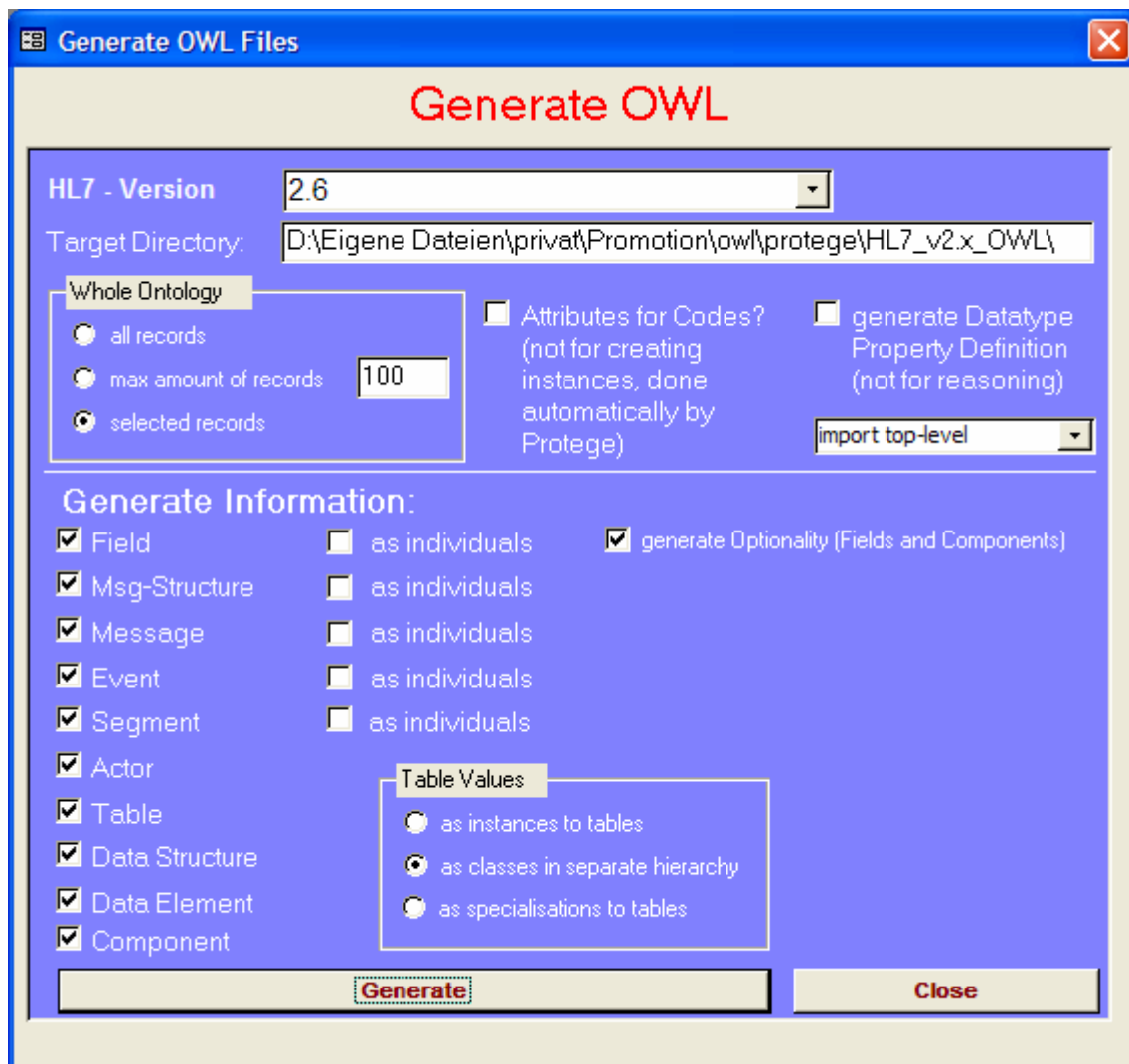


Abbildung 86: Formular zur Steuerung der HL7-v2.x-OWL-Generierung

Mit der dargestellten Selektion (selektierte Checkboxen und Radiobuttons) generiert das Programm die OWL-Dateien, wie sie in der Arbeit vorgestellt werden. Andere Selektionen erlauben andere Umsetzungen, bspw. als Instanzen (A-Box).

### 14.2.3. HL7 V3

Ein weiteres VBA-Programm erzeugt aus den MIF-Dateien die entsprechenden Ontologien. Auch hier wird nur ein Ausschnitt von V3 umgesetzt, der über die Ini-Datei entsprechend konfiguriert wird.

Das Formular zur Steuerung der Generierung der HL7 V3-Ontologie sieht im Prinzip ähnlich aus:

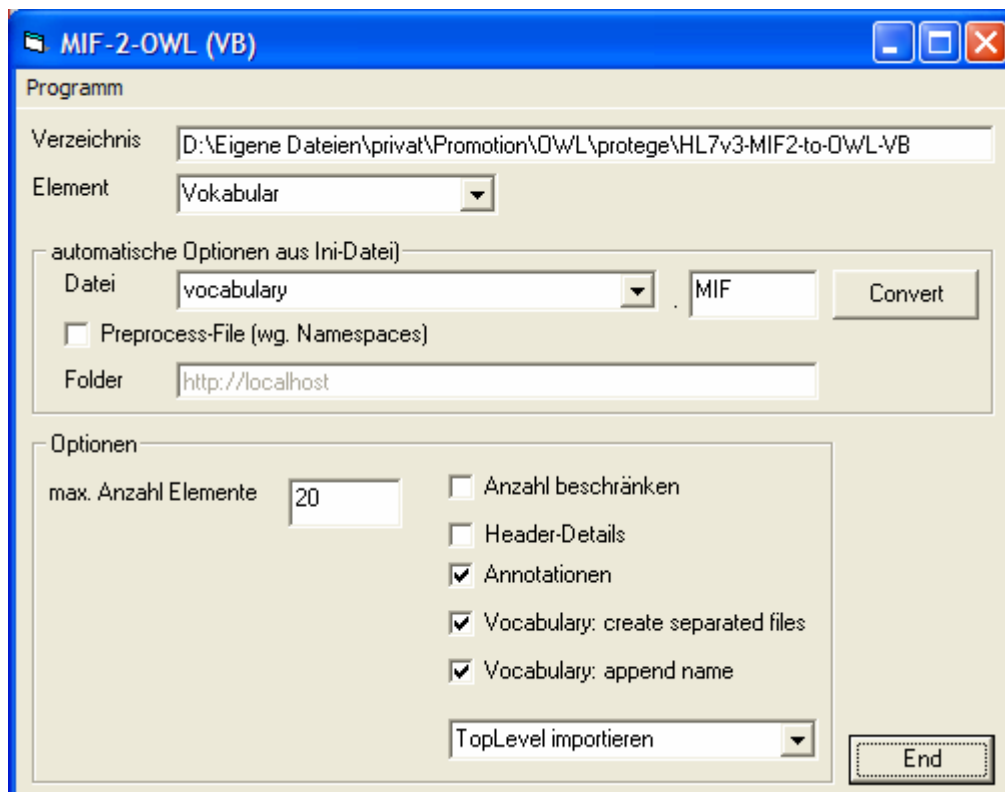


Abbildung 87: Formular zur Steuerung der HL7-V3-OWL-Generierung

#### 14.2.4. Import-Struktur und Namespaces

Eine der bereits erläuterten Schwierigkeiten besteht in dem Aufbau einer geeigneten Importstruktur, d.h. in der Festlegung, auf welchen Ontologien (OWL-Dateien) die jeweiligen Ontologien aufbauen. OWL unterstützt keine Includes<sup>43</sup> – d.h. Verteilung einer Ontologie über mehrere Dateien hinweg –, stattdessen muss jede einzelne Datei eine eigene Ontologie darstellen.

Umgekehrt kann genau diese Eigenschaft von OWL genutzt werden, um Namensgleichheiten bei Konzepten aufzulösen. So können bei HL7 V3 in den Vokabularien identische Namen für Vokabeldomänen (Vocabulary Domains), Codesysteme und Wertemengen (Value Sets) – beispielsweise "Administrative Gender" – genutzt werden. Durch die Aufteilung der Konzepte auf mehrere Dateien/Ontologien werden dann kein Konzeptpräfix (beispielsweise "cd\_" für "concept domain") notwendig, wenn stattdessen Namespaces eingesetzt werden. Durch diese Namespaces kann ontologieübergreifend eine eindeutige Identifikation der Konzepte erfolgen.

Umgekehrt kann über Prä- bzw. Postfixe ein eindeutiger Name erzeugt werden.

<sup>43</sup> Konventionelle Programmiersprachen stellen zwei verschiedene Mechanismen bereit, um mit verteilten Dateien umzugehen: „include“ und „import“. Das include-Statement zerlegt eine logische Datei in mehrere physikalische, während beim Import die einzelnen Dateien wieder in sich logisch und strukturell komplett sein müssen.

In dieser Arbeit wurde bei den V3-Vokabularen von Postfixen Gebrauch gemacht (vgl. Abbildung 43).

### **14.2.5. Namespaces und URLs zum Managen ununterscheidbarer Ontologie-Versionen**

Namespaces werden in Kombination mit einer URL aber auch zur eindeutigen Identifikation einer Ontologie genutzt. Eine URL stellt somit keine versionsunabhängige Kennung einer Ontologie dar.

Die Verwendung der Referenzontologie ACGT [ACGT] über eine gewisse Zeit hat gezeigt, dass dort Weiterentwicklungen<sup>44</sup> stattfinden. Sowohl bei BFO als auch ACGT unterliegen diese Weiterentwicklungen keiner Versionskontrolle – zumindest werden die einzelnen Versionen nicht mit einer eindeutigen Versionsnummer veröffentlicht. Die angegebenen Nummern verändern sich nicht, obwohl es im Zuge der Weiterentwicklung inkompatible Erweiterungen gegeben hat. Dies führt zu Importproblemen.

Eine Möglichkeit zur Lösung dieses Problems in Kombination mit der Loslösung von der Online-Charakteristik der genutzten Werkzeuge wie Protégé hat zur Entscheidung geführt, die Ontologien (BFO, ACGT) lokal im Zugriff zu haben und die URLs in den OWL-Dateien entsprechend abzuändern.

Eine Nutzung von Namespaces ist somit essentiell, da ansonsten nicht sauber mit den Konzepten hantiert werden kann.

### **14.2.6. Klasse vs. Instanz**

Ein Diskussionspunkt ganz anderer Art ist die Umsetzung eines Konzeptes als Klasse oder als Instanz in Kombination mit der Hinterlegung der Informationen. Wie in den bisherigen Ausführungen detailliert begründet, ist diese Entscheidung zu Gunsten von Klassen gefallen, da die Instanzen den konkreten Informationsobjekten vorbehalten sind.

### **14.2.7. Reasoning**

Das sogenannte Reasoning dient dazu, eine Ontologie auf Konsistenz zu überprüfen und die hinterlegten Regeln durchgängig anzuwenden sowie neue taxonomische Relationen zu errechnen.

Die anfangs verfügbaren Reasoner weisen hier aber gravierende Probleme auf. So ist Pellet 1.5 beispielsweise nicht in der Lage, in DatatypeProperties hinterlegte Infor-

---

<sup>44</sup> So wurde beispielsweise „Parent“ nicht mehr als Spezialisierung von „Organism“ (21.5.09) definiert, sondern mit „ParentRole“ als Spezialisierung von „RelativeRole“ eingeführt (13.6.10) – was eine Verbesserung darstellt und damit eigene Ergänzungsdefinitionen obsolet macht.

Ein weiteres nicht zu unterschätzendes Problem ist die fehlende inhaltliche Beschreibung der einzelnen Konzepte. Daher muss die Bedeutung aus der Hierarchie abgeleitet werden.

mationen korrekt zu behandeln, obwohl diese lediglich ignoriert werden müssten. Bei der Generierung einer Ontologie muss der geplante Einsatz dieses Reasoners insofern berücksichtigt werden, als dass entsprechende Detailinformationen dann nicht mit ausgegeben werden, um keine Fehler zu provozieren.

Bei Fact++ taucht dieses Problem nicht auf, so dass dieser als Reasoner favorisiert wird.

So führt beispielsweise die nachfolgende OWL-Definition zu Problemen:

```
<owl:Class rdf:about='&h17-v2;CNE.1'>
  <rdfs:subClassOf>
    <owl:Class rdf:about='&h17-v2;compCNE' />
  </rdfs:subClassOf>
  <rdfs:comment xml:lang="en">Identifizier</rdfs:comment>
  <rdfs:comment xml:lang="de">Schlüssel</rdfs:comment>
  </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource='#maxLength' />
      <owl:hasValue rdf:datatype='http://www.w3.org/2001/XMLSchema#int'>
        20
      </owl:hasValue>
    </owl:Restriction>
  </rdfs:subClassOf>
</rdfs:subClassOf>
...
</owl:Class>
```

Ohne DatatypeProperties	Mit DatatypeProperties
FaCT++ classified in 281ms	FaCT++ classified in 437ms org.semanticweb.owl.model.OWLRuntimeException: uk.ac.manchester.cs.factplusplus. owlapi.FaCTPlusPlusReasonerException: uk.ac.manchester.cs.factplusplus.FaCTPlusP lusException: FaCT++ Kernel: unsupported operation
Pellet classified in 672ms	Pellet classified in 688ms

Mit den aktuellen Versionen des Reasoners Pellet taucht dieses Problem nicht mehr auf. Es kommt „nur noch“ gelegentlich zu internen Warnungen.

### 14.2.8. Protégé 3.4 vs. 4.0.2 vs. 4.1

Es hat sich gezeigt, dass sich die beiden Programmversionen (3.4 und 4.0.2) insbesondere in der Funktionalität des Importes von Ontologien und der Darstellung der Properties unterscheiden. Die neuere Version scheint hier noch nicht so ausgereift zu sein, so dass je nach Fragestellung auf die besser passende Version zurückgegriffen wird. Im Juni 2010 ist die Version 4.1a erschienen, die aber nicht stabil ist und neue Fehler aufweist, so dass auf die Nutzung dieser Version verzichtet wird. Die seit August 2010 offiziell verfügbare Version 4.1β wird aber in dieser Arbeit aus Zeitgründen nicht mehr berücksichtigt.

Für beide Hauptversionen gilt, dass die zur Verfügung gestellte Editorfunktionalität nicht ausreicht, um bei Importhierarchien (geschachtelte OWL-Dateien) in Protégé vorgenommene Veränderungen auf die jeweilige Quelldatei zurückzuführen und dann dort entsprechend zu speichern. Daher kann Protégé für diese Arbeit grundsätzlich nur zur Visualisierung, nicht aber als Editor genutzt werden.

Auf die Probleme bei der Nutzung der verschiedenen Versionen von Protégé wird in den folgenden beiden Abschnitten genauer eingegangen.

#### **14.2.9. Fehler in Protégé 3.4**

Diese Version hat zwar eine weniger ausgereifte Benutzeroberfläche, dafür besteht hier die Möglichkeit, Namespaces mit anzuzeigen.

#### **14.2.10. Fehler in Protégé 4.0.2**

Die Grafiken und Screenshots in dieser Arbeit wurden mit Protégé 4.0.2 erstellt. Beim Einsatz traten jedoch diverse Probleme auf, die z.T. direkt auf Fehler von Protégé zurückgeführt werden können (In diesem Abschnitt wird "Ontologie" aufgrund der Import-Struktur synonym zu "OWL-Datei" verwendet):

##### Protégé Built 113:

- Es gibt Ladeprobleme, wenn die Quelldateien der Ontologien manuell aktualisiert werden: Protégé kann dann Änderungen teilweise nicht korrekt zuordnen. Nach der Aktualisierung fehlen dann manchmal Ontologien, die von der Änderung nicht betroffen sind. Abhilfe schafft hier nur das komplette Neuladen aller Ontologien d.h. OWL-Dateien.
- Das Repository registriert die Dateinamen der OWL-Dateien nicht korrekt. Problematisch ist das Laden über URLs, wenn kein Netzwerkzugang<sup>45</sup> besteht und aus diesem Grund der Netzwerkadapter heruntergefahren ist. Beim Laden muss dann umständlich jede einzelne Ontologie neu identifiziert werden.
- Anscheinend hat Protégé bei umfangreichen Ontologien Probleme mit Querbeziehungen, die auf gleichnamige Konzepte (Label) in unterschiedlichen Namespaces zugreifen. Bei Reduktion der Daten in den OWL-Dateien auf die konkret relevante Menge an Konzepten verändert sich zwar das Verhalten, aber es resultiert dennoch nicht in einem korrekten Ergebnis.

Protégé 3.4 besitzt die Funktionalität des Verfolgens von Verknüpfungen bei Selektion nicht.

##### Protégé Built 114:

- Diese Version erlaubt den korrekten Import von Ontologien, auch beim Update. Dafür müssen aber die URLs neu angepasst werden.

---

<sup>45</sup> Diese Problematik ist aber mitunter auch abhängig von der auf dem Rechner eingesetzten Firewall. Manche Internet Security Suites fahren bei nicht bestehendem Netzwerkzugang den Netzwerkadapter herunter, andere wiederum nicht.



- Es gibt Probleme mit dem Aktualisieren des Bildschirminhalts. Offensichtlich wird ein Refresh des Bildschirminhalts in einzelnen Bereichen in Abhängigkeit von der Mausposition und -bewegung nicht korrekt durchgeführt.
- Protégé visualisiert nicht alle eingegebenen bzw. generierten Informationen (beispielsweise gleichzeitiges Auftreten von minCardinality und maxCardinality). Die Editorfunktionalität von Protégé gestattet keine gleichzeitige Erfassung dieser Informationen. Die von W3C [W3C] herausgegebene OWL-Spezifikation erlaubt somit Konstrukte, die von Protégé (noch) nicht unterstützt werden. Umgekehrt werden diese Konstrukte in Protégé beim Laden von Ontologien nicht mit einem Fehler oder einer Warnung abgelehnt.

#### Protégé Built 115:

Hier ist in Bezug auf die für diese Arbeit relevante Funktionalität keine merkliche Veränderung gegenüber 114 festzustellen.

### **14.2.11. Festlegung eines konsistenten Subsets**

Die Nutzung von Ontologien setzt die Vollständigkeit<sup>46</sup> bzgl. eines bestimmten Szenarios voraus. Die einfachste Lösung für den Generierungsprozess wäre somit die Erzeugung der kompletten Ontologie. Dies führt aber automatisch zu einer unüberschaubaren Menge von Konzepten. Deshalb muss hier eine Reduktion durch Selektion stattfinden. Jedoch muss dazu ein Weg gefunden werden, wie die zu berücksichtigenden Elemente markiert werden können. Das Einfachste wäre eine Markierung in den Quelldateien. Dies führt nur dann nachhaltig zum Erfolg, wenn der Bearbeitende auch die Autorenschaft über die Quelldateien besitzt.

Bei der HL7-Datenbank sind diese Voraussetzungen gegeben. Daher kann hier ein zusätzliches Attribut (als weitere Spalte in den Tabellen) die Inklusion in die gewählte Menge anzeigen. Zusätzliche SQL-Abfragen können dann über eine Inkonsistenzsuche die Vollständigkeit sicherstellen. Alternativ kann nach der Generierung die Markierung für (nicht) definierte Klassen (zurück)gesetzt werden.

Bei den MIF-Dateien geht dies nur über externe Informationen. Bisher ist dies nur für die Wertemengen der Vokabularien notwendig, da deren Inhalte über sehr viele Dateien mit entsprechenden Namen verteilt sind. Diese Zusatzinformationen sind in der Konfigurationsdatei enthalten.

### **14.2.12. Implementierungsaspekte**

Darüber hinaus gibt es weitere Aspekte, die bei der Implementierung zu berücksichtigen sind. So spielt die Namenskonvention für Ontologien eine Rolle, da eindeutige Identifikatoren zur Konsistenzsicherung notwendig sind. Darüber hinaus sind

---

<sup>46</sup> Vollständig ist eine Menge von Klassen, wenn jede referenzierte Klasse auch definiert worden ist. Ansonsten kann Protégé nicht wissen, wo diese Klasse in der gesamten Hierarchie einzusortieren ist und ordnet diese dann unter "Thing" direkt ein.

Namespaces ausreichend zu definieren. In dieser Arbeit sind diese abgekürzt sowie mit Bindestrich und Kleinbuchstaben geschrieben worden. Für die Konzept-Namen und -Label sind ggf. Prä- und Postfixe festgelegt worden, um eine Einheitlichkeit sowie eine Reihenfolge bei der Sortierung zu gewährleisten.

Die Verzeichnis-Struktur der Dateien ist beim Import zwecks Generierung der entsprechenden Pfadangaben relevant.

Aber auch das Mapping selbst stellt gewisse Anforderungen. Bei der Namenskonvention sollte die Angabe der Version am Ende des Labels erfolgen, damit semantisch zusammengehörende Konzepte hintereinander stehen und so leichter zu finden sind. Bei den gemeinsamen Properties ist die aktuelle Ontologie-Struktur von Nachteil, da hier Constraints aufwändiger werden. Durch Einziehen einer Zwischenebene, damit nicht alle Einträge flach untereinander stehen, würde die Verwaltung deutlich vereinfacht, da zusätzliche Gruppierungen möglich sind, die dann ausgeblendet werden. Dies ist derzeit nur für die Tabellen/Codesysteme gemacht worden.

In OWL selbst bestehen ebenfalls Einschränkungen hinsichtlich der Nutzung der bereit gestellten Konstrukte: OWL differenziert zwischen "rdf:about", "rdf:ID" und "rdf:resource", die unterschiedlich gehandhabt werden müssen: Die Nutzung in Kombination mit bzw. ohne Raute ("#") bzw. Namespace-Angabe führt dann zu unterschiedlichen Ergebnissen. Ebenso kann die Einschränkung (Constraints) von Object-Properties (Property-Hierarchie) entweder explizit/allgemein oder speziell in den Klassen erfolgen.

#### **14.2.13. Eine kleine Statistik**

Ein weiteres Ergebnis dieser Arbeit offenbart sich in der nachfolgenden Übersicht, in der typische Eigenschaften der OWL-Dateien aufgeführt sind. In Anbetracht der Importstruktur (s.o.) addieren sich natürlich die Werte. Ein paar Zahlen sind bemerkenswert und deshalb in Tabelle 21 schattiert dargestellt:

Tabelle 21: Eine kleine Statistik der Klassen und Beziehungen

Ontologie	Anz. Klassen	Anz. Object Property	Anz. Data Property	Anz. Individuals	Anz. Subclass Axiome	Anz. Equivalent Classes Axioms	Ausdrucksstärke <sup>47</sup>
RO	0	23	0	0	0	0	ALHI+
Protégé DC	0	0	1	15	0	0	AL(D)
BFO	39	0	1	1	38	13	ALC(D)
ACGT	1790	228	16	62	2438	67	SROIQ(D)
ACGT Addendum	1868	228	18	78	2515	69	SROIQ(D)
CSO	95	47	5	17	97	18	SHIQ(D)
CSO Addendum	120	53	8	18	143	18	SHIQ(D)
HL7 v3	3513	335	5	95	7091	21	SHIQ(D)
Mapping V3	5393	551	25	159	9661	77	SROIQ(D)
HL7 v2.6	892	65	10	17	2504	18	SHIQ(D)
Mapping v2.6	2786	283	30	81	5144	74	SROIQ(D)

Die Relationship Ontology (RO) besteht ausschließlich aus ObjectProperties und ist für den Import in andere Ontologien konzipiert. BFO selbst definiert nur eine äußerst geringe Anzahl von Klassen (39). CSO als Referenzontologie führt ebenfalls nur einige wenige Klassen ein (95), so dass sich dies harmonisch in die Grundphilosophie der OBO-Foundry einfügt.

<sup>47</sup> Die Ausdruckstärke gibt die Mächtigkeit der verwendeten Axiome an. Im Einzelnen stehen die Buchstaben für: [Grim2009, Horr2010]

ALC = kleinste geschlossene präpositionale Beschreibungslogik

H = Rollenhierarchien

S = ALC mit transitiven Rollen

R = Rollenoperationen

O = Nominals

I = inverse Rollen

F = funktionale Rollen

Q = qualifizierte Zahlenrestriktionen

N = Zahlenrestriktionen

(D) = Datentypen

Erst die darauf aufbauenden Ontologien (ACGT, HL7 V3, HL7 v2.x) addieren eine bemerkenswerte Anzahl von Klassen (1790, 3513, 892). Selbst bei einer Beschränkung auf einen kleinen überschaubaren Bereich bei HL7 v2.x und V3, ist eine deutliche Anzahl von Klassen hinzugefügt worden. Insofern lässt sich leicht mutmaßen, was bei einer Ausdehnung auf die gesamte Informationsmenge der Kommunikationsstandards passieren würde.

## 15. Bereitstellung der Ergebnisse

Damit die in dieser Arbeit entwickelten Ontologien in anderen Arbeiten verwendet werden können, werden diese online zur Verfügung gestellt. Die URLs hierzu lauten:

<http://www.oemig.de/frank/phd-thesis.html>

Unter dieser URL sind auch die ausgelagerten Anhänge zu finden.

### 15.1. Ontologien

Der Download der OWL-basierten Ontologien umfasst folgende Dateien:

Tabelle 22: Dateiliste der entwickelten Ontologien

Dateiname	Inhalt	Unterverzeichnis
CSO.owl	Top-Level-Ontologie der Kommunikationsstandards	BFO
CSO-Addendum.owl	Zusatz zu CSO für das Mapping	
acgt-fo.owl	Add-Ons für ACGT	
hl7_v2.6.owl	HL7 v2.6	HL7_v2.x_OWL
RIM.owl	Reference Information Model	HL7v3-MIF2-to-OWL-VB
CMET.owl	Common Message Element Types	
DataType.owl	Data Types	
HL7-V3.owl		
Vocabulary.owl	Vocabulary (enthält nur Imports)	
vocab_codesystem.owl	Codesysteme	
vocab_conceptDomain.owl	Concept Domains	
vocab_valueSet.owl	Value Sets	
PRPA_IN201301UV02.owl	Interaktionen (Beispiele)	
PRPA_IN201302UV02.owl		
PRPA_IN201303UV02.owl		
PRPA_IN402001UV02.owl		
PRPA_MT201301UV02.owl	Message Types (Beispiele)	
PRPA_MT201302UV02.owl		

Dateiname	Inhalt	Unterverzeichnis
PRPA_MT201303UV02.owl		
PRPA_MT402001UV02.owl		
Mapping Master_v2.6.owl	Mapping für v2.x - ACGT/BFO	Mapping
Mapping Master_V3.owl	Mapping für V3 – ACGT/BFO	

Die Dateien sind in das entsprechende Unterverzeichnis<sup>48</sup> unter <http://localhost/owl/protege> einzustellen.

Das gilt auch für die benutzten Ontologien (BFO, etc.). Wie bereits erwähnt, befinden sich diese ebenfalls in einem separaten Entwicklungsprozess, so dass sie möglicherweise asynchron überarbeitet werden und dann nicht mehr mit den in dieser Arbeit entwickelten Ontologien verwendet werden können. Diese sind ebenfalls in das Verzeichnis BFO einzustellen:

Tabelle 23: Dateiliste der benutzten BFO-Ontologien

Dateiname	Inhalt	Ursprung	Datum <sup>49</sup>
ro.owl	Reference Ontology	<a href="http://www.ifomis.org/obo/ro/1.0">http://www.ifomis.org/obo/ro/1.0</a>	13.06.10
bfo-1.1.owl	Basic Formal Ontology	<a href="http://www.ifomis.org/acgt/1.0">http://www.ifomis.org/acgt/1.0</a>	13.06.10
acgt-1.0.owl	Advanced Clinico Genomic Trial Ontology	<a href="http://www.ifomis.org/bfo/1.1">http://www.ifomis.org/bfo/1.1</a>	13.06.10
protege-dc.owl	DCMI Namespace for the Dublin Core Metadata Element Set, Version 1.1	<a href="http://purl.org/dc/elements/1.1/">http://purl.org/dc/elements/1.1/</a>	13.06.10
iao.owl	Information Artefact Ontology	<a href="http://purl.obolibrary.org/obo/iao.owl">http://purl.obolibrary.org/obo/iao.owl</a>	

In den hier bereitgestellten Dateien wurden der Dateiname und die jeweilige Referenzen angepasst, so dass sie in Protégé über <http://localhost> geladen und genutzt werden können. Dies ermöglicht, unabhängig von den Quelldaten (offline) zu arbeiten. Außerdem ist die resultierende Entkopplung von den nicht-versionierten Updates bei BFO und ACGT von Vorteil. Inhaltlich wurde nichts geändert.

<sup>48</sup> Bei den meisten Webservern (beispielsweise Apache) können die Verzeichnisse konfiguriert/gemappt werden, so dass diese Festlegung kein Problem darstellen sollte.

<sup>49</sup> Aufgrund der bereits erwähnten Versionierungsproblematik bei OBO wurde ein bestimmter Stand „eingefroren“ und lokal genutzt. Zum Abschluss der Arbeit wurden die verwendeten OWL-Dateien aktualisiert, allerdings konnte hier dann kein essentieller Unterschied mehr festgestellt werden.

## **15.2. Ausgelagerte Anhänge**

Die textuellen Anhänge zu dieser Arbeit sind einerseits zu umfangreich, andererseits nur bedingt für das Verständnis der Schrift relevant. Daher werden sie nicht direkt als Anhang eingebunden und stattdessen ebenfalls auf der Homepage bereitgestellt. Die Inhalte strukturieren sich wie folgt:

### 1. Standards als Grundlage der Arbeit

- 1.1. EDI
- 1.2 ASC X.12
- 1.3 HL7
- 1.4 SCIPHOX
- 1.5 Detailed Clinical Models

### 2. Detaileinsichten

- 2.1 HL7 Version 2.x
  - 2.1.1 Datentypen
  - 2.1.2 Ereignisse
  - 2.1.3 dynamisches Verhalten
  - 2.1.4 Löschanforderungen
  - 2.1.5 Null-Flavors
  - 2.1.6 Übertragungsprotokolle
  - 2.1.7 Profile
- 2.2 HL7 Version 3
  - 2.2.1 Zustandsübergänge
  - 2.2.2 Mood-code ("Modus")
  - 2.2.3 Anwendungsrollen
  - 2.2.4 Datentypen
  - 2.2.5 ITS
  - 2.2.6 „Structural Attributes“
  - 2.2.7 „mandatory Values“
  - 2.2.8 „Null-Flavors“
  - 2.2.9 Publishing

### 3. Verbesserung der Interoperabilität

- 3.1. v2.x-Publishing
- 3.2. Die HL7-Datenbank
- 3.3. Alternativen zum Füllen der Datenbank mit Werten
- 3.4. Dokumentation der Schnittstelle
- 3.5. V3-Publishing
- 3.6. Dokumentenorientierte Spezifikationen
- 3.7. v2.x: Komponentenmodell für Datentypen
- 3.8. v2.x: implizite Segmentstrukturen
- 3.9. v2.x: explizite Nachrichtenstrukturen/BNF

- 3.10. v2.x: Nationale Weiterentwicklung
- 3.11. V3: Nationale Weiterentwicklung
- 3.12. v2.x: XML Encoding
- 3.13. v2.x: Nachrichtenprofile
- 3.14. v2.x: Kompatibilität zwischen den Versionen
- 3.15. Zeichensatz
- 3.16. V3: Lokalisation
- 3.17. CMET-Substitution
- 3.18. V3: Interversion-Compatibility
- 3.19. V3: Mixtur aus Versionen
- 3.20. CDA
- 3.21. Konformanztests

## E. Diskussion

### 16. Charakterisierung der entwickelten Lösung

Die vorliegende Arbeit bietet eine wissenschaftlich fundierte und zugleich praktische Lösung für die Koordinierung (Mapping) verschiedener HL7-Standards an. Wie in der Herleitung dargestellt, unterscheidet sich der hier vorgestellte Ansatz fundamental von den existierenden Lösungen. Die meisten konkurrierenden Mapping-Ansätze verwenden die XML-Schemas von Nachrichteninstanzen. Dabei werden die einzelnen Elemente mit einer Programmiersprache aufeinander abgebildet. Zum Teil geschieht dies mit algorithmischer Unterstützung.

Der Ansatz von Bicer [Artemis, Bice2005] benutzt zwar OWL zur Ontologie-Repräsentation, greift aber ebenfalls auf Nachrichteninstanzen zurück.

Die anderen OWL-basierten Ansätze für HL7 beschränken sich auf vereinzelte Artefakte von HL7 V3 in jeweils unterschiedlichen Ausprägungen – allerdings ohne den Anspruch irgendeiner Integration. Zusammenfassend werden in den konkurrierenden Ansätzen weder Ontologien formal abgeleitet noch gemappt.

Der in der Dissertation entwickelte Ansatz leitet über das benutzte Framework des GCM, das eine systemorientierte und architekturzentrierte Modellierung der Realität anbietet und so auch Ontologien und ihr Design einschließt, für unterschiedliche Kommunikationsstandards einheitliche Grundlagen ab und baut anschließend systematisch darauf auf, um ein Mapping zu unterstützen.

Durch den Mediatoransatz können prinzipiell gleiche Kommunikationsstandards wie die verschiedenen Ausprägungen einer Familie direkt aufeinander gemappt werden. Unterschiedliche Standards müssen jedoch nur einmal auf die Fachdomäne und nicht immer wieder direkt aufeinander abgebildet werden.

Außerdem werden hierbei durch den komponentenbasierten Ansatz des GCM gleiche Fragmente wiederverwendet (beispielsweise Name für Patienten, Ärzte, Angehörige).



Im Ergebnis basiert diese Arbeit auf einer echten Formalisierung der Kommunikationsstandards.

## 17. Bewertung des Nutzens

### 17.1. Bewertung der eigenen Arbeit

Die Arbeit trägt wesentlich zur Sicherung der praktischen Interoperabilität von Informationssystemen im Gesundheitswesen bei, die über Kommunikationsstandards verbunden sind. Dabei werden entscheidende Schwächen konkurrierender Ansätze überwunden, wie die fehlende Berücksichtigung der wichtigen architekturellen Aspekte. Die Lösung basiert auf Ontologien und somit formal repräsentiertem Wissen. In der nachfolgenden Tabelle werden die in dieser Arbeit ausgearbeiteten Ergebnisse der Situation vorher im Einzelnen gegenübergestellt:

Tabelle 24: Bewertung der eigenen Arbeit (vorher/nachher)

Vorher	Nachher
Indirekt vorhandenes nicht formalisiertes Wissen	Definition von Wissen und Wissensrepräsentation im praktischen Kontext der Arbeit unter Bezug auf das GCM
divergierendes Verständnis unter den Experten, was die einzelnen Begriffssysteme ausmacht bzw. unterscheidet	Analyse der Eigenschaften der Begriffssysteme mit anschließender Modellierung über UML-Klassendiagramme und Einordnung in das GCM zur Klärung der Unterschiede
vereinfachtes Meta-Modell der ersten Kommunikationsstandards der HL7-v2.x-Familie	Analyse der Kommunikationsstandards mit Erstellung entsprechender UML-Klassendiagramme; anschließende Einordnung in das GCM sowie Ableitung der daraus folgenden Klassenhierarchie
Das HL7 V3 RIM als Grundlage für Modelle (D-MIM, ..)	Einordnung von HL7 V3 in das GCM mit Aufzeigen der fehlenden Elemente
-	Darstellung der Kommunikationsstandards mit Hilfe formaler Ontologien auf Basis von BFO
-	Ausarbeitung einer Architektur für die geforderte Wissensbasis unter Bezugnahme auf eine gemeinsame Referenzontologie; letztere wird in der Arbeit als CSO bezeichnet
-	Formalisierung der Kommunikationsstandards: HL7 v2.x: Generierungsprozess aus der HL7-Datenbank heraus HL7 v3: Generierung aus den MIF-Files (RIM, Datentypen, Vocabulary, ausgewählte Domänen)

Entwicklung einer ontologiebasierten Architektur zur Sicherung semantischer Interoperabilität zwischen Kommunikationsstandards im Gesundheitswesen

Vorher	Nachher
-	Visualisierung der Wissensbasis über ein Werkzeug (Protégé)
-	Analyse der Anforderungen an ein Mapping-Verfahren zur Überbrückung zwischen den Kommunikationsstandards
-	Erarbeitung eines gemeinsamen Referenzmodells; hier Überarbeitung vorhandener Referenzontologien mit Auflistung der Schwachstellen. Die Entwicklung der Top-Level-Ontologie hat mehrere Redesigns durchlaufen, um die Gemeinsamkeiten von HL7 v2.x und V3 zu erforschen <ul style="list-style-type: none"> <li>• gemeinsame Top-Level-Konzepte</li> <li>• gemeinsame Property-Struktur</li> </ul>
-	Ergänzung bestehender Ontologien (ACGT) um weitere Details (ACGT-FO)
-	Aufdeckung einiger Probleme mit ACGT wie die falsche Modellierung der Konzepte (beispielsweise IdentifierPart)
-	Die Property-Hierarchie in BFO und ACGT ist sehr flach, obwohl eine Strukturierung in Bezug auf die gemeinsame Top-Level-Ontologie, die dann auch anders aussehen kann, sinnvoll wäre; Überschneidungen sind schon vorhanden

Eine Bewertung lässt sich aber auch über die Unterteilung in Vor- und Nachteile durchführen, wobei die beiden hier nachfolgenden Listen keine direkte Gegenüberstellung darstellen:

Tabelle 25: Bewertung der eigenen Arbeit (Vor-/Nachteile)

Pro (Vorteile)	Contra (Nachteile)
<ul style="list-style-type: none"> <li>• Feingranularer konzeptbasierter Ansatz</li> <li>• Formalisierung der Kommunikationsstandards</li> <li>• Wiederverwendung des Mappings über Nachrichtengrenzen hinweg</li> <li>• Mediatoransatz</li> </ul>	<ul style="list-style-type: none"> <li>• Ein manuelles Mapping ohne Werkzeugunterstützung ist sehr mühsam, aber auch nicht intendiert.</li> <li>• Rechenintensive Auswertung</li> </ul>

---

## **17.2. Zukünftige Nutzung der Ergebnisse**

Die in dieser Arbeit erarbeitete Formalisierung der Kommunikationsstandards mit einem Mapping auf eine ebenfalls formalisierte Fachdomäne präzisiert das Wissen über die ausgetauschten Inhalte. Damit könnte beispielsweise die Konfiguration von Kommunikationsservern automatisiert werden.

Eine ganz andere Einsatzmöglichkeit wäre die Nutzung der hinterlegten Informationen in einer ontologiegetriebenen Architektur (ODA). Allerdings müssen dann Anforderungen an die Performance definiert werden, da schon der hier präsentierte kleine Ausschnitt relativ viele Konzepte enthält, die über ihre Beziehungen ausgewertet werden müssen.

## **17.3. Übertragung auf der Ergebnisse auf Persistenzstrukturen**

Aus Schnittstellensicht gibt es drei Möglichkeiten, Informationen zu persistieren, d.h. dauerhaft zu speichern. Für Business-Objekte und Datenbankstrukturen werden die Meta-Modelle nachfolgend aufgelistet. Auf Dokumente und deren mögliche Strukturen und Formate wird nicht näher eingegangen, da sie prinzipiell ähnlich zum Nachrichtenparadigma zu behandeln sind. Interessant wären hier nur Formate, die die Inhalte für eine automatisierte maschinelle Verarbeitung zur Verfügung stellen, da sonst Interoperabilität nur durch die Interpretation von Texten über sprachverarbeitende Verfahren ermöglicht werden kann. Für CDA Rel.2 wurde schon in Abschnitt 6.4 auf die Möglichkeit der Nutzung der hier entwickelten Verfahren hingewiesen. Daher kann die Analyse der Persistenzstrukturen auf Business-Objekte bzw. auf Fachdomäneninhalte reduziert werden.

### **17.3.1. UML-Modell für Business-Objekte**

Ein Business-Objekt ist im Wesentlichen ein extern ansprechbares Objekt, das Parameter oder direktmanipulierbare Attribute sowie Methoden anbietet, die diese (auch internen) Attribute über eine festgelegte Logik verändern oder abfragen. Dies entspricht dem klassischen CORBA-Ansatz [CORBA] der Object Management Group [OMG].

Die Attribute oder Parameter müssen dann wiederum einen bestimmten Datentyp haben, der wie in den anderen Modellen aufgebaut sein kann.

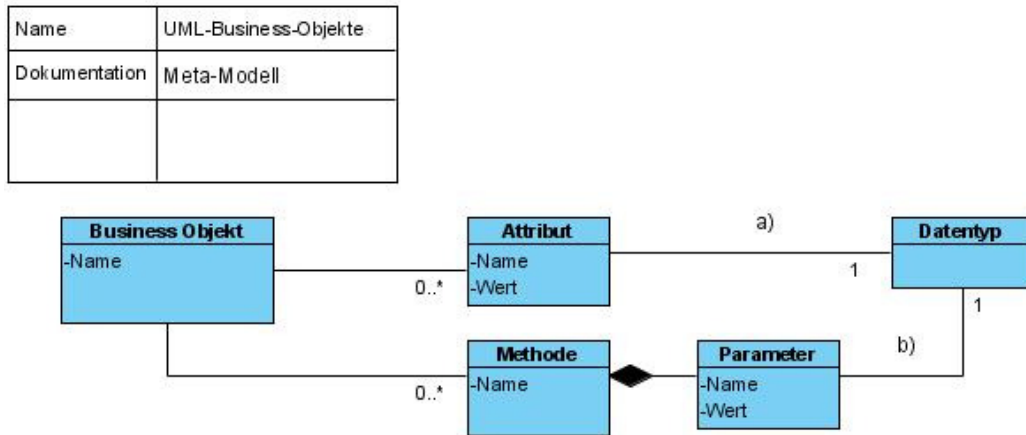


Abbildung 88: UML-Business-Objekte

Dieses UML-Modell richtet sich nach den Basis-Konzepten für Komponentenarchitekturen, die Blobel in [Blob2000] näher ausführt. Im Prinzip stellt dieses Diagramm eine Verfeinerung von "Service" und "Interface" nach den GCM-Prinzipien dar, so dass die in dieser Arbeit entwickelten Mechanismen in Bezug auf die feingranularen Konzepte anwendbar werden.

Für die in Abbildung 88 dargestellten Objekte und die sie repräsentierenden Klassen gilt:

- a) Attribute besitzen eine Datenstruktur.
- b) Parameter: Hier gilt im Prinzip das Gleiche wie für die Attribute. Allerdings werden die Daten in einer XML-Struktur mittels Web-Service Description Language (WSDL, [WSDL]) übertragen.

### 17.3.2. UML-Modell für relationale Datenbanken

Das UML-Modell für die Datenbank soll folgende drei Möglichkeiten der Datenspeicherung berücksichtigen:

- statisches Modell (feste Tabellen)
- dynamisches Modell (generierbare Tabellen mit frei definierbarer Struktur)
- generisches Modell (Tabellen gemäß Attribut-Wert-Zuordnung)

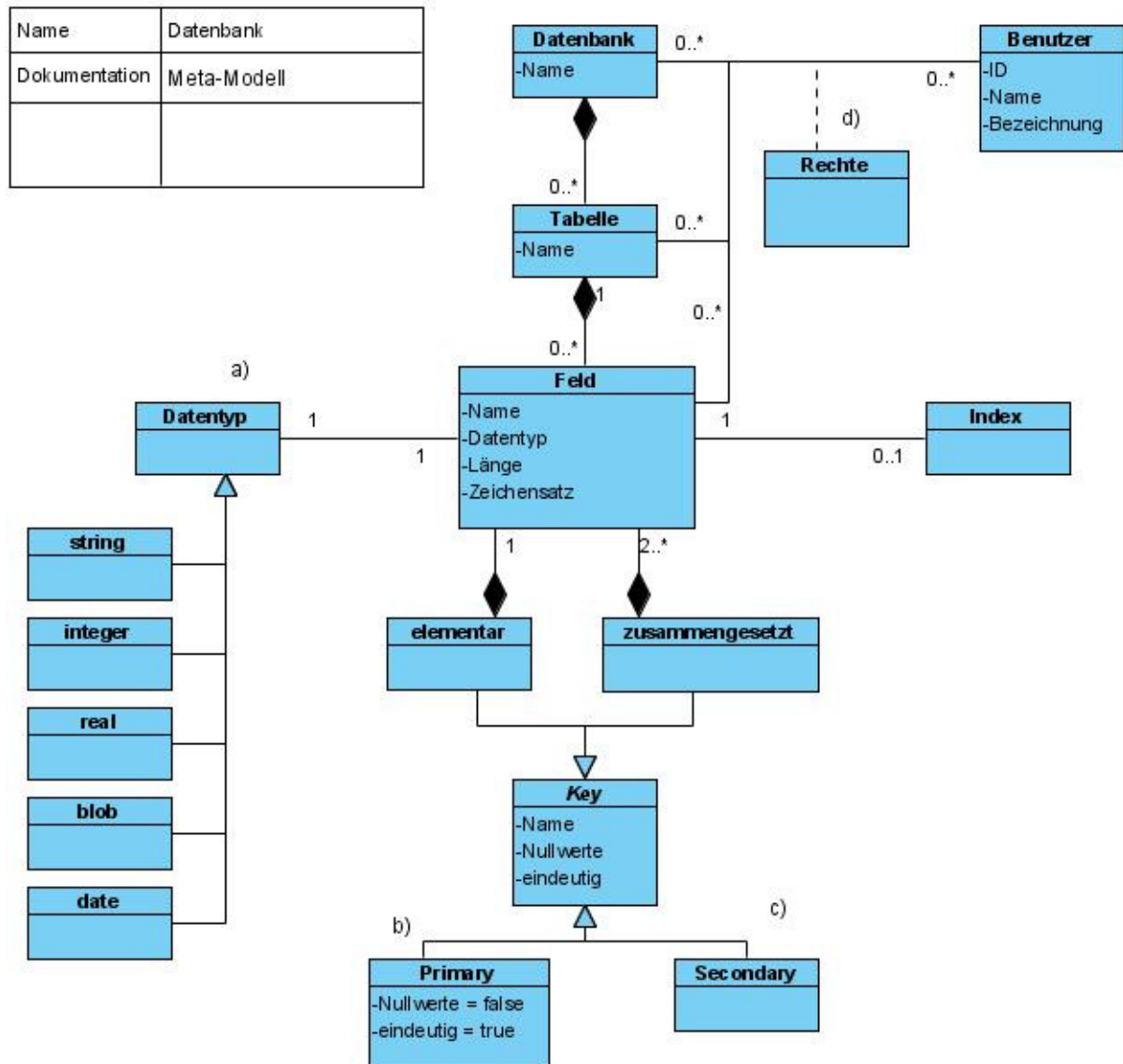


Abbildung 89: UML-Datenbankmodell

Für die Klassen im UML-Datenbankmodell (Abbildung 89) gilt.

- a) Der Datentyp muss nicht unbedingt separat modelliert werden.
- b) Der Primary-Key bezieht sich auf eine Tabelle. Er kann dort aus mehreren Feldern bestehen, muss jedoch eine Eindeutigkeit herstellen.
- c) Der Secondary-Key (Fremdschlüssel) kann mehrere Felder zusammenfassen, die in anderen Tabellen den Primärschlüssel darstellen.
- d) Rechte können vererbt werden.

### 17.3.3. Abbildungsprozess

Praktische Erfahrungen aus diversen Projekten haben gezeigt, dass für die Übertragung persistenter Daten über einen Kommunikationsstandard kein zusätzlicher Zwischenschritt durch Einsatz eines weiteren Kommunikationsstandards erfolgen sollte. Damit lässt sich die hier notwendige Arbeit auf folgende Szenarien reduzieren:

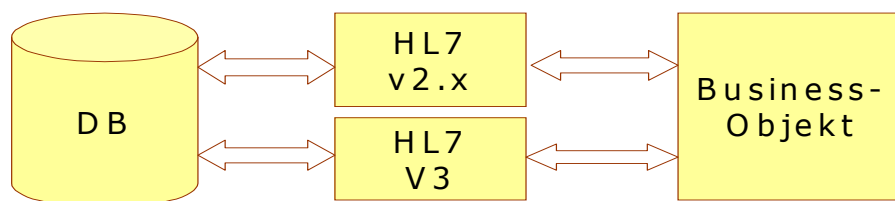


Abbildung 90: Umsetzungsprozess

Als weiteres Ergebnis lässt sich somit festhalten, dass die in dieser Arbeit entwickelten Prinzipien auch auf Persistenzstrukturen übertragbar sind. Es wäre „lediglich“ die Aufbereitung einer separaten Struktur unter BFO mit Einsatz des Mappings durchzuführen. Ein derartiger Ansatz schöpft die Möglichkeiten des GCM noch besser aus.

## 18. offene Punkte

Durch die Fokussierung der Arbeit auf die Erzeugung der Ontologien und die Einordnung in das GCM bleibt eine Reihe von Punkten, die nachträglich noch weiter evaluiert werden können:

- Constraints: Einige der Informationsmodelle in HL7 (insbesondere die D-MIMs in HL7 V3) enthalten Randbedingungen/Einschränkungen, die in natürlicher Sprache formuliert sind. Es gilt, diese dann sowohl in OCL als auch in OWL formal auszudrücken.
- Editor: Die direkten Ontologien werden wie aufgezeigt generiert. Das dazugehörige Mapping wird aber derzeit direkt in OWL mit einem Texteditor erarbeitet. Hier wird ein integrierter Editor benötigt, der eine geeignete Funktionalität mit Zugriff auf die Konzepte anbietet. Dies könnte über ein Plug-In für Protégé realisiert werden. Hierbei wäre gleichzeitig ein geeigneter Exportmechanismus zu realisieren, da Protégé eine Aufteilung der Ontologien in verschiedene Dateien nicht vornehmen kann.
- Erarbeitung weiterer Details, d.h. die Integration weiterer Nachrichtenspezifikationen inkl. des dazugehörigen Mappings in die Fachdomäne.
- Pflege der Wissensbasis: die erarbeitete Ontologie und das dazugehörige Mapping müssen weiter ausgebaut werden.
- Weiterentwicklung in Bezug auf IHE: In dem IHE Whitepaper [IHE Vol.0] wurde demonstriert, dass die einzelnen Frameworks gerade in Bezug auf das Konzept der Optionalität (R, RE, O, ..) nicht korrekt aufgebaut sind.
- Diskussion der Ergänzungen zu ACGT mit den Autoren von ACGT und BFO.
- Einbringung von CSO in die OBO-Foundry.
- Anwendung auf Nachrichteninstanzen: Erarbeitung einer Anwendung, die einen Parser und Generatoren für Nachrichteninstanzen enthält.

- 
- Roundtrip in der Generierung: Überprüfung der Korrektheit des Mappings durch ein Mapping des Ergebnisses zurück auf die Eingangsinformationen, d.h. es müsste dann die Eingangsnachricht wieder entstehen.
  - Update auf die neueste Version, d.h. Nutzung der neuesten HL7-Spezifikationen als auch Nutzung weiterer Details aus den neuesten MIF-Versionen.
  - Angleichung der eingeführten Relationen mit RO.
  - Übertragung auf andere Nachrichtenstandards wie beispielsweise DICOM und xDT.
  - Diskussion der Arbeit in Referenz zu neuesten Entwicklungen an der Service Oriented Architecture (SOA) Front einschließlich der Bemühungen einer ontologischen Repräsentation dieser Entwicklungen (z.B. The Open Group's Service-Oriented Architecture Ontology [TOG2009] oder OMG's Ontology Definition Metamodel).

### **18.1. Ausblick auf zukünftige Arbeiten**

Ein Teil der offenen Punkte kann sehr gut über separate Folgearbeiten wie beispielsweise Diplomarbeiten aufgelöst werden. Vorschläge für solche Aktivitäten sind:

- Implementierung eines Tools, das das Mapping unterstützt, da die manuelle Definition und Erfassung sehr mühsam ist.
- Implementierung eines Agenten, der das Mapping zur Laufzeit auswertet und auf einen Datensatz anwendet.
- Ausarbeitung weiterer Domänenontologien für andere Kommunikationsstandards wie beispielsweise DICOM und xDT.
- Realisierung des Mappings auf die bereits erwähnten Persistenzstrukturen.
- Performanceauswertungen, d.h. eignet sich das erarbeitete Vorgehen für den Einsatz in der Praxis?

### **18.2. Architekturskizze der Anwendung**

Die OWL-Dateien werden derzeit in Protégé geladen und dort zur Anzeige gebracht. Protégé ist dabei eine Anwendung, die auf der Eclipse Runtime Plattform basiert und ein API für die Bearbeitung der Daten zur Verfügung stellt. Auf dieser Basis können Plug-Ins entwickelt werden, die eine erweiterte Verarbeitung zulassen [KnuFer2004]. Der OWL-Property-Wizzard [OWLPropViz, Wach2008] ist eine derartige Erweiterung.

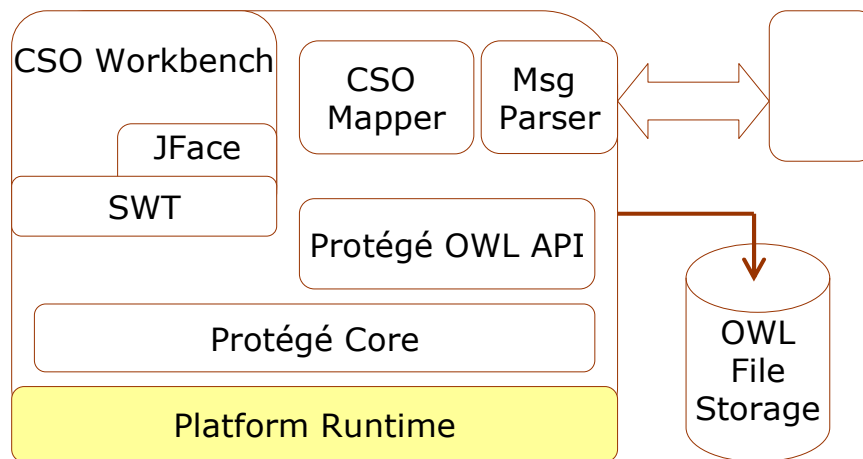


Abbildung 91: Architekturskizze

Umgekehrt müssten die Protégé Core Komponenten dann ebenfalls dank der OSGi Technologie in einer selbstentwickelten Anwendung genutzt werden können, um in Kombination mit anderen Komponenten (Nachrichten-Parser und Generator) konkrete Nachrichteninstanzen entsprechend mappen zu können [Künn2009]. Eine derartige Anwendung müsste dann auch als Service auf einer Windows\_Plattform lauffähig sein.

## 19. Zusammenfassung und Schlussfolgerungen

Die Forderung nach ausgebauter Kommunikation und Kooperation im Gesundheitswesen ist verbunden mit der Forderung nach der (semantischen oder working) Interoperabilität der Anwendungen. Allein durch den Druck der Anwender wird die Interoperabilität verbessert werden – nicht zuletzt auch durch die Abnahme proprietärer Formate. Letztere sind in der Entwicklung und in der Pflege zu teuer. Um diese Anforderungen erfüllen zu können, sind Anpassungen der Systemarchitektur unumgänglich.

Nach einer umfassenden Aufbereitung der Grundlagen und Klärung der Begrifflichkeiten wird die dieser Arbeit zugrunde liegende Methodik – das generische Komponentenmodell (GCM) – begründet und erläutert. Das Ziel dieser Arbeit, die Entwicklung einer ontologiebasierten Architektur zur Sicherung semantischer Interoperabilität zwischen Kommunikationsstandards im Gesundheitswesen, wird über einen Mediatoransatz erreicht, der eine Fachdomäne als Brücke zwischen kommunizierenden Domänen in den Mittelpunkt stellt. Der Einsatz des GCMs als Rahmenarchitektur unterstützt dabei nicht nur die Analyse der notwendigen Komponenten, sondern auch den Aufbau und die Einhaltung der Architektur, die in der OWL-Darstellung über die Beziehungen zwischen den Konzepten zum Ausdruck kommt. Sie setzt damit die Erkenntnis über die fundamentale Rolle der Architektur in Informationssystemen um [Blob2009b]. Außerdem erlaubt der Architekturansatz die Erweiterung der Methodologie hin zu ontologie-getriebenen Lösungen auf der Basis von Services über das Kommunikationsparadigma hinaus. Damit zeichnet es zukünftige Entwicklungen des HL7-Standards vor.



Die hierarchische Einbindung einer gemeinsamen Top-Level Ontologie wie BFO sichert dabei die Konsistenz der Ontologien.

Im Vergleich zu bereits existierenden Werken beschreitet diese Arbeit gänzlich neue Wege, indem versucht wird, anhand zweier Kommunikationsstandards aufzuzeigen, wie über die Entwicklung geeigneter Meta-Modelle Wissen in konzeptualisierter Form dargestellt und mit Hilfe eines Mediators für Mappings genutzt werden kann. Durch diese Indirektion lassen sich prinzipiell beliebige Standards auf einander abbilden, da ein einmal definiertes Mapping durch den feingranularen Ansatz wiederverwendet werden kann. Ähnliche Standards – beispielsweise aus derselben Familie - können sogar direkt gemappt werden. Über das gewählte Werkzeug wurde demonstriert, wie das Mapping dann in der Praxis funktioniert.

Bisherige Arbeiten haben keinen derartigen Anspruch und Umfang.

## F. Anhänge

### 20. Anhang A: Einführung in die Grundlagen

In der Arbeit werden eine Reihe von Standards genutzt, die hier kurz vorgestellt werden sollen. Für ein tiefergehendes Verständnis sei auf die jeweilige Literatur verwiesen.

#### ***20.1. RDF: Resource Description Framework***

Mit der zunehmenden Verbreitung des Internets stellt sich die Frage nach den Möglichkeiten, wie eine bestimmte Information (wieder- bzw. auf-)gefunden werden kann? Das elementarste Vorgehen ist hierbei, als Suchergebnis alle Seiten aufzulisten, die alle Worte oder Phrasen enthalten, die als Suchanfrage verwendet wurden.

Wenn diese Worte auf der Seite gleichzeitig als Schlüsselwörter deklariert sind, so lässt sich auf deren Relevanz schließen und die Seite entsprechend höher bewerten und somit weiter vorne im Suchergebnis einsortieren. Die Suchmaschinenanbieter [Google, Yahoo] haben hier ihre eigene Methodik entwickelt, die besten Ergebnisse ausfindig zu machen.

Schwierig wird es, wenn zweideutige Wörter – wie beispielsweise "Essen" oder "Bank" - benutzt werden. Dann lässt es sich nicht mehr so einfach auf die Bedeutung schließen. So ist bei "Essen" beispielsweise unklar, ob die Stadt oder eine Mahlzeit gemeint ist.

Des Weiteren gibt es im Internet eine Vielzahl an Seiten zu verwandten Themen. In den allermeisten Fällen stehen diese Seiten ohne Beziehung zueinander.

In beiden Fällen ist es wünschenswert, über eine zusätzliche Auszeichnung der Seiten Anfragen in einer höheren Qualität zu beantworten bzw. überhaupt Antworten auf vorher unbeantwortete Fragen zu erhalten. Diese Idee ist die Grundlage des Semantic Web, in dem der "AAA-Slogan" gilt: "Anyone can say Anything about Any topic" [AllHen2008].

Die Basis bildet hierbei die Möglichkeit, einfache Statements über die einzelnen Seiten abzugeben. Hierbei muss jedoch berücksichtigt werden, dass diese Statements nicht eindeutig, unvollständig, widersprüchlich und zeitlich eingeschränkt sind.

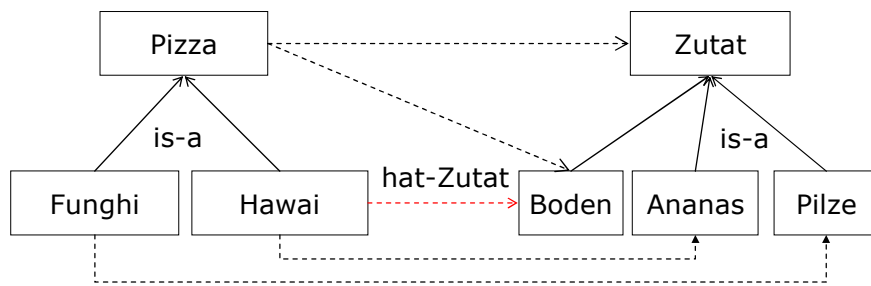


Abbildung 92: Beispiel für einfache semantische Aussagen

Wenn jemand die Aussagen gemäß vorhergehender Abbildung (Kästchen und Pfeile) trifft, so kann ein anderer daraus eine Schlussfolgerung ziehen, die als rot gestrichelter Pfeil dargestellt ist.

Das Resource Description Framework [RDF] liefert die Grundlage, um die dazu notwendigen Informationen formalisiert ausdrücken zu können. Dabei werden die Informationen auf einfache Aussagen gemäß der folgenden Form reduziert:

Subjekt Prädikat Objekt

Das obige Beispiel würde demnach wie folgt dargestellt:

Tabelle 26: RDF-Statements

Subjekt	Prädikat	Objekt	Bemerkung
Funghi	is-a	Pizza	
Hawai	is-a	Pizza	
Boden	is-a	Zutat	
Ananas	is-a	Zutat	
Salami	is-a	Zutat	
Pizza	hat-Zutat	Zutat	
Funghi	hat-Zutat	Pilze	
Hawai	hat-Zutat	Ananas	
Pizza	hat-Zutat	Boden	
Hawai	hat-Zutat	Boden	<= abgeleitet

Diese Aussagen können dann noch mit Quantoren versehen werden, die in obiger Grafik nicht eingezeichnet sind. So lässt sich beispielsweise ausdrücken, dass jede Pizza einen Boden hat, also auch die "Pizza Hawai".

Umgekehrt lassen sich anders herum aber auch Schlussfolgerungen ableiten, wenn man beispielsweise weiß dass eine Pizza Pilze enthält, dann ist es eine "Funghi". (Das ist jetzt eine stark vereinfachte Aussage, da sich auf diese Weise sehr komplexe Sachverhalte darstellen lassen, die nicht ohne weiteres zu übersehen sind.)

## 20.2. OWL: Web Ontology Language

Aufbauend auf RDF vereinheitlicht die Web Ontology Language OWL [OWL, AllHen2008] eine Reihe weiterer Konstrukte, die für eine Bezugnahme auf die Semantik notwendig sind, so dass Anwendungen es damit einfacher haben. So wird obiges Pizza-Beispiel wie folgt dargestellt:

```
<?xml version="1.0"?>

<!DOCTYPE rdf:RDF [
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY owl2xml "http://www.w3.org/2006/12/owl2-xml#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY pizza "http://localhost/owl/protege/pizza.owl#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>

<rdf:RDF xmlns="http://localhost/owl/protege/pizza.owl#"
  xml:base="http://localhost/owl/protege/pizza.owl"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
  xmlns:pizza="http://localhost/owl/protege/pizza.owl#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  <owl:Ontology rdf:about="" />

  <!--
  ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
  //
  // Object Properties
  //
  ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
  -->

  <!-- http://localhost/owl/protege/pizza.owl#hasZutat -->
  <owl:ObjectProperty rdf:about="#hasZutat">
    <rdfs:domain rdf:resource="#Pizza"/>
    <rdfs:range rdf:resource="#Zutat"/>
  </owl:ObjectProperty>

  <!--
  ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
  //
  // Classes
  //
  ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
  -->

  <!-- http://localhost/owl/protege/pizza.owl#Ananas -->
  <owl:Class rdf:about="#Ananas">
    <rdfs:subClassOf rdf:resource="#Zutat"/>
  </owl:Class>
```

```

<!-- http://localhost/owl/protege/pizza.owl#Boden -->
<owl:Class rdf:about="#Boden">
  <rdfs:subClassOf rdf:resource="#Zutat"/>
</owl:Class>

<!-- http://localhost/owl/protege/pizza.owl#Funghi -->
<owl:Class rdf:about="#Funghi">
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasZutat"/>
      <owl:onClass rdf:resource="#Pilze"/>
      <owl:qualifiedCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:qualifiedCardinality>
    </owl:Restriction>
  </owl:equivalentClass>
  <rdfs:subClassOf rdf:resource="#Pizza"/>
</owl:Class>

<!-- http://localhost/owl/protege/pizza.owl#Hawai -->
<owl:Class rdf:about="#Hawai">
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasZutat"/>
      <owl:onClass rdf:resource="#Ananas"/>
      <owl:qualifiedCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:qualifiedCardinality>
    </owl:Restriction>
  </owl:equivalentClass>
  <rdfs:subClassOf rdf:resource="#Pizza"/>
</owl:Class>

<!-- http://localhost/owl/protege/pizza.owl#Pilze -->
<owl:Class rdf:about="#Pilze">
  <rdfs:subClassOf rdf:resource="#Zutat"/>
</owl:Class>

<!-- http://localhost/owl/protege/pizza.owl#Pizza -->
<owl:Class rdf:about="#Pizza">
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasZutat"/>
      <owl:onClass rdf:resource="#Boden"/>
      <owl:qualifiedCardinality
rdf:datatype="&xsd;nonNegativeInteger">1</owl:qualifiedCardinality>
    </owl:Restriction>
  </owl:equivalentClass>
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasZutat"/>
      <owl:someValuesFrom rdf:resource="#Zutat"/>
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>

<!-- http://localhost/owl/protege/pizza.owl#Tomate -->
<owl:Class rdf:about="#Tomate">
  <rdfs:subClassOf rdf:resource="#Zutat"/>
</owl:Class>

<!-- http://localhost/owl/protege/pizza.owl#Zutat -->
<owl:Class rdf:about="#Zutat"/>
</rdf:RDF>

```

## In Protégé:

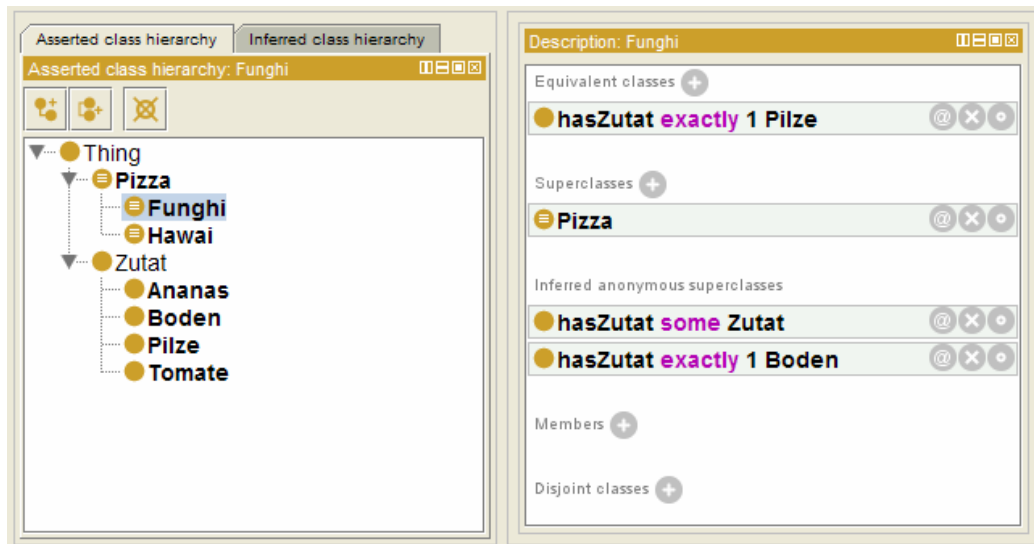


Abbildung 93: OWL-Pizza-Beispiel in Protégé

### 20.2.1. Die Untersprachen von OWL 1

OWL in der Version 1.0 wird in drei verschiedenen Untersprachen zum Einsatz gebracht, die eine unterschiedliche Mächtigkeit in der Ausdrucksfähigkeit besitzen:

**OWL Lite:** Diese Sprache stellt nur wenige Konstrukte für einfache Klassifikationen zur Verfügung. Dazu gehören auch eingeschränkte Aussagen zur Kardinalität.

OWL Lite ist eine Untermenge von OWL DL.

**OWL DL:** Die DL-Variante – DL steht für Description Logic – umfaßt relativ viele Konstrukte, die aber noch ein Reasoning und Vollständigkeitsprüfungen erlauben.

OWL DL ist eine Untermenge von OWL Full.

**OWL Full:** Die maximale Ausdrucksfähigkeit von OWL wird mit dieser Untersprache erreicht. So ist es beispielsweise möglich auszudrücken, dass eine Klasse gleichzeitig die Unterklasse zu einer anderen sowie die Instanz einer dritten Klasse ist.

### 20.2.2. Die Sprachkonstrukte

Nachfolgend die Liste der Sprachkonstrukte in OWL Lite: [OWL]

Tabelle 27: OWL1 Sprachkonstrukte

RDF Schema Features:	(In)Equality:	Property Characteristics:	Property Restrictions:
Class (Thing, Nothing) rdfs:subClassOf rdf:Property rdfs:subPropertyOf rdfs:domain rdfs:range Individual	equivalentClass equivalentProperty sameAs differentFrom AllDifferent distinctMembers	ObjectProperty DatatypeProperty inverseOf TransitiveProperty SymmetricProperty FunctionalProperty InverseFunctional-	Restriction onProperty allValuesFrom someValuesFrom

		Property	
Restricted Cardinality:	Header Information:	Class Intersection:	
minCardinality (only 0 or 1) maxCardinality (only 0 or 1) cardinality (only 0 or 1)	Ontology imports	intersectionOf	
Versioning:	Annotation Properties:	Datatypes	
versionInfo priorVersion backwardCompatibleWith incompatibleWith DeprecatedClass DeprecatedProperty	rdfs:label rdfs:comment rdfs:seeAlso rdfs:isDefinedBy AnnotationProperty OntologyProperty	xsd datatypes	

OWL DL wird um folgende Konstrukte gegenüber OWL Lite erweitert:

Tabelle 28: OWL DL Sprachkonstrukte

Class Axioms:	Boolean Combinations of Class Expressions:	Arbitrary Cardinality:	Filler Information:
oneOf, dataRange disjointWith equivalentClass (applied to class expressions) rdfs:subClassOf (applied to class expressions)	unionOf complementOf intersectionOf	minCardinality maxCardinality cardinality	hasValue

### 20.2.3. OWL 2

OWL ist für viele Aufgaben noch nicht ausreichend. Mit OWL 2.0 sollte aufgrund der Praxiserfahrungen mit OWL 1.0 eine zusätzliche Ausdruckstärke durch neue ontologische Axiome, nicht-logische Erweiterungen sowie eine Überarbeitung der OWL-Varianten hergestellt werden. Ziel war es aber, die Kompatibilität zum existierenden OWL-Standard zu erhalten. Ein weiterer wichtiger Punkt war hierbei die Aufrechterhaltung der Entscheidbarkeit [OWL2, HitKrö2008].

Als Beispiele sei hier angeführt [HitKrö2008]:

OWL 1:

Person  $\sqcap \geq 3$  hatKind  
"Klasse aller Personen mit 3 oder mehr Kindern"

OWL 2:

Person  $\sqcap \geq 3$  hatKind.(Frau  $\sqcap$  Professor)  
"Klasse aller Personen mit 3 oder mehr Töchtern, die Professorinnen sind"

Ein anderes Beispiel, das in SHOIN (die Erläuterung zur Ausdrucksstärke findet sich in Tabelle 21 auf Seite 159) ausgedrückt werden kann, ist die Transitivität von „hatFreund“:

„Die Freunde meiner Freunde sind auch meine Freunde.“

In SHOIN kann aber nicht ausgedrückt werden.

„Die Feinde meiner Freunde sind auch meine Feinde.“

Nachfolgend die wichtigsten Erweiterungen, die in OWL2 definiert sind [OWL2]:

- Neue Konstrukte für Properties
- Erweiterte XSD Datentypnutzung
- Top und Bottom Properties

Darüber hinaus sind damit drei Sprachprofile festgelegt worden [OWL2 Profile]:

- OWL 2 EL
- OWL 2 QL
- OWL 2 RL

#### **20.2.4. SWRL**

Die *“Semantic Web Rule Language”* ist ein Vorschlag für eine Regelsprache, die auf OWL aufbaut, um Regeln darstellen zu können. So kann nachfolgendes Beispiel nicht direkt in OWL ausgedrückt werden [SWRL]:

$$\text{hasParent}(?x,?y) \wedge \text{hasBrother}(?y,?z) \Rightarrow \text{hasUncle}(?x,?z)$$

Die Lesart folgt dem Prinzip, dass wenn die Vorbedingung gilt, auch die Schlussfolgerung gültig ist.

### **20.3. OPPL: *Ontology Preprocessor Language***

OPPL ist vereinfacht ausgedrückt eine Regelsprache, um OWL-basierte Ontologien zu manipulieren. Für ein korrektes Reasoning sollten die zugrunde liegenden Ontologien vollständig sein, d.h. alle Statements enthalten. Wie in der Arbeit dargestellt, werden die OWL-Dateien aus einer Datenbank bzw. den XML-basierten HL7-V3-MIF-Dateien generiert. Während dieses Generierungsprozesses ist es mitunter nur schwer realisierbar, dass alle Statements erzeugt werden. Daher bietet sich hier OPPL an, um die fehlenden Statements nachträglich zu ergänzen. Dies betrifft insbesondere die vollständige Definition von disjunkten Klassen. (Es kann davon ausgegangen werden, dass alle Geschwisterklassen zueinander diskunkt sind.) Dies lässt sich am einfachsten über OPPL-Regeln erzeugen. [OPPL]



## 21. Anhang B: OWL-Spezifikationen

Eine Liste der erarbeiteten Spezifikationen findet sich auf Seite 160 als Bereitstellung der Ergebnisse und wird deshalb hier nicht noch einmal aufgelistet. Dieses Kapitel beschäftigt sich stattdessen mit der Ausführung der Details aus der MIF2-Spezifikation:

### 21.1. Umsetzung von HL7 V3 MIF nach OWL

Die MIF2-Dateien unterscheiden sich gravierend hinsichtlich ihres Inhalts. Mangels einer brauchbaren Dokumentation bleibt nur das Reverse-Engineering, um die Inhalte nutzbar zu machen:

#### 21.1.1. RIM

Die Klassen des Referenz-Informations-Modells von HL7 V3 sind in der Datei „RIM.coremif“ enthalten:

##### 21.1.1.1. MIF-Datei

Tabelle 29: MIF-Struktur für Elemente des RIMs

RIM-Klassen	Kommentar
<code>&lt;?xml version="1.0" encoding="UTF-8"?&gt;</code>	XML-Header
<code>&lt;staticModel   xmlns:xsi=     "http://www.w3.org/2001/XMLSchema-instance"   xmlns="urn:hl7-org:v3/mif"   name="0219" title="HL7 Reference Information Model"   representationKind="flat"   isSerializable="false"   xsi:noNamespaceSchemaLocation=     "..\\schemas\\mifStaticModelFlat.xsd"&gt;</code>	
<code>&lt;historyItem dateTime="3/21/2007" version="02-19"&gt; &lt;/historyItem&gt;</code>	
<code>&lt;packageLocation   root="DEFN" realm="Universal" artifact="RIM"/&gt;</code>	
<code>&lt;header&gt;</code>	
<code>&lt;renderingInformation schemaVersion="1"   renderingTime="3-22-2008T24:01:25"   application="RoseTree 4.2.7"&gt;   &lt;renderingNotes&gt; &lt;/renderingNotes&gt; &lt;/renderingInformation&gt;</code>	Rendering-Informationen, u.a. zur Aufbereitung der Grafik
<code>&lt;legalese copyrightOwner="Health Level Seven"   copyrightYears="2008"&gt;</code>	

RIM-Klassen	Kommentar
<code>&lt;notation&gt;All rights reserved.&lt;/notation&gt;</code> <code>&lt;/legalese&gt;</code>	
<code>&lt;responsibleGroup</code> <code>    organizationName="Health Level Seven"/&gt;</code>	
<code>&lt;contributor&gt;</code> <code>    &lt;role/&gt;</code> <code>    &lt;name key="McKenzie, Lloyd"</code> <code>        name="Lloyd McKenzie"/&gt;</code> <code>    &lt;affiliation/&gt;</code> <code>&lt;/contributor&gt;</code>	Liste der Beitragenden
....	
<code>&lt;/header&gt;</code>	
<code>&lt;annotations&gt;</code> .... <code>&lt;/annotations&gt;</code>	
<code>&lt;figure&gt;</code> <code>    &lt;fixedImage&gt;</code> <code>        &lt;pixmap href="./Graphics/RIM_billboard.pdf"/&gt;</code> <code>    &lt;/fixedImage&gt;</code> <code>&lt;/figure&gt;</code>	Informationen über die Grafikdatei
<code>&lt;ownedSubjectAreaPackage</code> <code>    name="FoundationClasses" sortKey="1"&gt;</code>	Aufteilung der Gesamtinformationen in best. Bereiche
<code>&lt;header&gt;</code>	
<code>    &lt;ballotInfo</code> <code>        ballotStatus=" Normative Ballot"</code> <code>        ballotOccurrence="1"/&gt;</code>	
<code>&lt;/header&gt;</code>	
<code>&lt;figure&gt;</code> <code>    &lt;fixedImage&gt;</code> <code>        &lt;pixmap</code> <code>            href="./Graphics/RIM_NormativeContent.gif"/&gt;</code> <code>    &lt;/fixedImage&gt;</code> <code>&lt;/figure&gt;</code>	
<code>&lt;annotations&gt;</code> .... <code>&lt;/annotations&gt;</code>	
<code>&lt;ownedSubjectAreaPackage</code> <code>    name="Acts" sortKey="1"&gt;</code>	
....	

RIM-Klassen	Kommentar
<code>&lt;/ownedSubjectAreaPackage&gt;</code>	
<code>&lt;ownedSubjectAreaPackage     name="Entities" sortKey="2"&gt;</code>	Bereich der "Entities"
<code>&lt;header&gt;     &lt;ballotInfo         ballotStatus="Normative Ballot"         ballotOccurrence="1"/&gt; &lt;/header&gt;</code>	
<code>&lt;figure&gt;     &lt;fixedImage&gt;         &lt;pixmap href="./Graphics/RIM_Entities.gif"/&gt;     &lt;/fixedImage&gt; &lt;/figure&gt;</code>	
<code>&lt;annotations&gt;     ..... &lt;/annotations&gt;</code>	
<code>&lt;ownedClass name="Entity"/&gt; &lt;ownedClass name="Place"/&gt;</code>	Angabe der dazugehörigen Klassen
<code>    .....</code>	
<code>&lt;/ownedSubjectAreaPackage&gt;</code>	
<code>    ....</code>	
<code>&lt;/ownedSubjectAreaPackage&gt;</code>	
<code>&lt;ownedEntryPoint className="Entity"/&gt; &lt;ownedEntryPoint className="Act"/&gt;</code>	Liste der Entry-Points
<code>    ....</code>	
<code>&lt;ownedClass&gt;</code>	Detaillierte Informationen zu den Klassen
<code>&lt;class name="Person" isAbstract="false"&gt;</code>	Name der Klasse
<code>&lt;supplierStructuralDomain mnemonic="PSN"/&gt;</code>	
<code>&lt;annotations&gt;     ..... &lt;/annotations&gt;</code>	Anmerkungen zur Klasse
<code>&lt;attribute name="addr" sortKey="1"     minimumMultiplicity="0"     maximumMultiplicity="*"     isMandatory="false"&gt;</code>	Attribute-Header
<code>&lt;annotations&gt;     .... &lt;/annotations&gt;</code>	Anmerkungen zum Attribut
<code>&lt;type name="BAG"&gt;</code>	Datentypangabe mit

RIM-Klassen	Kommentar
<pre> &lt;supplierBindingArgumentDatatype   name="AD"/&gt; &lt;/type&gt; </pre>	Vocabulary-Domain
<pre> &lt;/attribute&gt; </pre>	
<pre> &lt;attribute name="maritalStatusCode" &gt;   ... &lt;/attribute&gt; </pre>	Weitere Attribute
<pre> ... </pre>	
<pre> &lt;/class&gt; </pre>	
<pre> &lt;/ownedClass&gt; </pre>	
<pre> ..... </pre>	Weitere Klassen
<pre> &lt;ownedAssociation&gt; </pre>	Relationen zwischen den Klassen
<pre> &lt;annotations&gt;   .... &lt;/annotations&gt; </pre>	
<pre> &lt;connections&gt; </pre>	
<pre> &lt;traversableConnection   name="languageCommunication"   minimumMultiplicity="1"   maximumMultiplicity="1"   participantClassName="LanguageCommunication"   sortKey="C"/&gt; </pre>	Die beiden Richtungen der Relation
<pre> &lt;traversableConnection   name="entity"   minimumMultiplicity="0"   maximumMultiplicity="*"   participantClassName="Entity" sortKey="Z"/&gt; </pre>	
<pre> &lt;/connections&gt; </pre>	
<pre> &lt;/ownedAssociation&gt; </pre>	
<pre> ... </pre>	Weitere Relationen
<pre> &lt;/staticModel&gt; </pre>	

### 21.1.1.2. Ontologie-Struktur

Tabelle 30: Ontologie-Struktur für das RIM

MessageElement	Kommentar
RIM-Part	Oberklasse für Details der RMIMs
RIM-Class	

<RIM-Klasse>	dedizierte RIM-Klassen; diese haben teilweise noch eine hierarchische Struktur
RIM-Attribute	Attribute zu den RIMs: separate Hierarchie mit Querverweisen
<RIM-Klasse>-Attr	
<RIM>.<Attribut>	
RIM-Association	Realisierung der Relation zwischen den RIM-Klassen
<RIM>-<Association>	

### 21.1.1.3. importierte Ontologien

- Vocabulary

### 21.1.2. Vocabulary

#### 21.1.2.1. MIF-Datei

Tabelle 31: MIF-Struktur für Vocabulary

Ausschnitt aus vocabulary.mif	Kommentar
<?xml version="1.0" encoding="UTF-8"?>	XML-Header
<pre>&lt;vocabularyModel   name="498-20080320"   title="HL7 Vocabulary"   packageKind="version"   definitionKind="partial-publishing"   schemaVersion="2.0"   xmlns:x="http://www.w3.org/1999/xhtml"&gt;</pre>	
<pre>&lt;packageLocation   root="DEFN"   artifact="VO"   realmNamespace="UV"   version="498-20080320"/&gt;</pre>	
<pre>&lt;header&gt;   &lt;renderingInformation     renderingTime="2008-03- 22T12:02:47"     application="RoseTree 4.2.7"/&gt;   &lt;legalese     copyrightOwner="HL7, Inc."     copyrightYears="2008"/&gt; &lt;/header&gt;</pre>	
<pre>&lt;conceptDomain   name="AcknowledgementCondition"&gt;   &lt;annotations/documentation/definition/text&gt;     ...   &lt;/annotations/documentation/definition/text</pre>	Umsetzung als eigene Ontologie: Jeder Eintrag ist eine eigene Klasse.

Ausschnitt aus vocabulary.mif	Kommentar
<pre>&gt;   &lt;/conceptDomain&gt;</pre>	
<pre>...</pre>	Infos über weitere Konzepte
<pre>&lt;codeSystem   name="AcknowledgementCondition"   title="AcknowledgementCondition"   packageKind="name"   codeSystemId="2.16.840.1.113883.5.1050"&gt;</pre>	Umsetzung als eigene Ontologie: Jeder Eintrag ist eine eigene Klasse. Der Klassenname wird dabei aus dem Attribut "name" gebildet.
<pre>&lt;header&gt; ... &lt;/header&gt;</pre>	
<pre>&lt;annotations&gt; ... &lt;/annotations&gt;</pre>	
<pre>&lt;releasedVersion releaseDate="2008-03-20"   publisherVersionId="29"   hl7MaintainedIndicator="true"   completeCodesIndicator="true"   hl7ApprovedIndicator="true"&gt;</pre>	
<pre>&lt;supportedConceptRelationship   relationshipKind="Specializes"   name="Specializes"   inverseName="Generalizes"   isNavigable="true"   reflexivity="irreflexive"   symmetry="antisymmetric"   transitivity="transitive"&gt;   &lt;description&gt;     ...   &lt;/description&gt; &lt;/supportedConceptRelationship&gt;</pre>	
<pre>&lt;supportedConceptRelationship   relationshipKind="Generalizes"   name="Generalizes"   inverseName="Specializes"   isNavigable="true"   reflexivity="irreflexive"   symmetry="antisymmetric"   transitivity="transitive"&gt;   &lt;description&gt;     ...   &lt;/description&gt; &lt;/supportedConceptRelationship&gt;</pre>	
<pre>&lt;supportedConceptProperty   propertyName="internalId"   type="Token"   isMandatoryIndicator="false"&gt;   &lt;description&gt;     ...   &lt;/description&gt;</pre>	

	Ausschnitt aus vocabulary.mif	Kommentar
	<pre data-bbox="533 255 943 282">&lt;/supportedConceptProperty&gt;</pre>	
	<pre data-bbox="533 293 1038 555">&lt;supportedConceptProperty   propertyName="status"   type="Token"    isMandatoryIndicator="false"&gt;   &lt;description&gt;     ...   &lt;/description&gt; &lt;/supportedConceptProperty&gt;</pre>	
	<pre data-bbox="533 562 847 651">&lt;supportedLanguage&gt; en &lt;/supportedLanguage&gt;</pre>	
	<pre data-bbox="533 663 1023 1106">&lt;concept&gt;   &lt;annotations&gt;     ...   &lt;/annotations&gt;   &lt;conceptProperty     name="internalId"     value="10126"/&gt;   &lt;printName language="en"     preferredForLanguage="true"     text="Always"/&gt;   &lt;code     code="AL"     status="active"/&gt; &lt;/concept&gt;</pre>	<p data-bbox="1090 663 1410 913">Jedes Konzept bildet einen eigenen Eintrag. Hierbei bildet der Code in Kombination mit dem codesystem einen eigenen Eintrag.</p>
	<pre data-bbox="533 1117 1023 1561">&lt;concept&gt;   &lt;annotations&gt;     ...   &lt;/annotations&gt;   &lt;conceptProperty     name="internalId"     value="10128"/&gt;   &lt;printName language="en"     preferredForLanguage="true"     text="Error/reject only"/&gt;   &lt;code     code="ER"     status="active"/&gt; &lt;/concept&gt;</pre>	
	<pre data-bbox="533 1572 1023 2016">&lt;concept&gt;   &lt;annotations&gt;     ...   &lt;/annotations&gt;   &lt;conceptProperty     name="internalId"     value="10127"/&gt;   &lt;printName language="en"     preferredForLanguage="true"     text="Never"/&gt;   &lt;code code="NE"     status="active"/&gt; &lt;/concept&gt;</pre>	
	<pre data-bbox="533 2027 679 2060">&lt;concept&gt;</pre>	

	Ausschnitt aus vocabulary.mif	Kommentar
	<pre> &lt;annotations&gt; ... &lt;/annotations&gt; &lt;conceptProperty   name="internalId"   value="10129"/&gt; &lt;conceptProperty   name="status"   value="retired"/&gt; &lt;printName language="en" preferredForLanguage="true" text="Successful only"/&gt; &lt;code code="SU"   status="retired"/&gt; &lt;/concept&gt; </pre>	
	<pre> &lt;/releasedVersion&gt; </pre>	
	<pre> &lt;/codeSystem&gt; </pre>	
<pre> ... </pre>		<p>Infos über weitere Code-Systeme</p>
	<pre> &lt;valueSet   sortKey="AcknowledgementCondition"   id="2.16.840.1.113883.1.11.155"   name="AcknowledgementCondition"&gt;   &lt;annotations&gt;   ...   &lt;/annotations&gt;   &lt;version versionDate="2008-03-20"&gt;     &lt;supportedCodeSystem&gt;       2.16.840.1.113883.5.1050     &lt;/supportedCodeSystem&gt;     &lt;supportedLanguage&gt;       en     &lt;/supportedLanguage&gt;     &lt;content       codeSystem="2.16.840.1.113883.5.1050"/&gt;     &lt;/version&gt;   &lt;/valueSet&gt; </pre>	<p>Umsetzung als eigene Ontologie: Jeder Eintrag ist eine eigene Klasse.</p> <p>Der Abschnitt "content" definiert auf unterschiedliche Art, welche Codes bzw. Codesysteme zu kombinieren sind.</p>
<pre> ... </pre>		<p>Infos über weitere Value-Sets</p>
	<pre> &lt;contextBinding   conceptDomain="AcknowledgementCondition"   valueSet="2.16.840.1.113883.1.11.155"   bindingRealmName="undetermined"   effectiveDate="2008-03-20"/&gt; </pre>	
<pre> ... </pre>		<p>Infos über weitere Kontext-Bindings</p>
	<pre> &lt;/vocabularyModel&gt; </pre>	



### 21.1.2.2. Ontologie-Struktur

Tabelle 32: Ontologie-Struktur für Vocabulary

MessageElement	Beschreibung
Vocabulary	Oberklasse für Details der RMIMs
conceptDomain	
<Konzept-Domäne>	
codeSystem	
<Codesystem>	
<Codesystem>.<value>	
valueSet	
<ValueSet>	

### 21.1.3. R-MIM: MIF2-Struktur

#### 21.1.3.1. MIF-Datei

Im Folgenden wird die MIF-Struktur für R-MIMs wiedergegeben.

Tabelle 33: MIF-Struktur für R-MIM

Ausschnitt aus PRPA_MT201301UV.mif	Kommentar
<code>&lt;?xml version="1.0" encoding="UTF-8"?&gt;</code>	XML-Header
<pre> &lt;mif:staticModel   xmlns:mif="urn:hl7-org:v3/mif"   xmlns="urn:hl7-org:v3/mif"   xmlns:xhtml="http://www.w3.org/1999/xhtml"   xmlns:xsi="     "http://www.w3.org/2001/XMLSchema-instance"   isAbstract="false"   isSerializable="true"   name="UV"   packageKind="realmNamespace"   representationKind="flat"   schemaVersion="2.0"   xsi:schemaLocation=""&gt; </pre>	Root-Element
<pre> &lt;mif:packageLocation   artifact="MT-deprecated"   domain="PA"   id="201301"   realmNamespace="UV"   root="DEFN"   section="AM"   subsection="PR"/&gt; </pre>	Meta-Daten

Entwicklung einer ontologiebasierten Architektur zur Sicherung semantischer Interoperabilität zwischen Kommunikationsstandards im Gesundheitswesen

Ausschnitt aus PRPA_MT201301UV.mif	Kommentar
<code>&lt;mif:header&gt;</code>	Header
<code>&lt;mif:renderingInformation application="Visio to MIF transform"/&gt;</code>	
<code>&lt;mif:legalese copyrightOwner="Health Level Seven" copyrightYears="2008"/&gt;</code>	mit Copyright
<code>&lt;/mif:header&gt;</code>	
<code>&lt;mif:graphicRepresentation presentation="HL7"&gt;</code>	
<code>&lt;mif:graphElement name="Main"&gt; &lt;mif:position x="0" y="0"/&gt; &lt;mif:size height="1343" width="1672"/&gt; &lt;/mif:graphElement&gt;</code>	
<code>&lt;entryPoint name="Patient Activate"&gt; &lt;mif:graphElement   containerDiagramName="Main"   shapeId="Sheet.139"&gt;   &lt;mif:position x="689" y="2"/&gt;   &lt;mif:size height="69" width="230"/&gt;   &lt;mif:anchorage     connectToShapeId="Sheet.139"&gt;     &lt;mif:position x="823" y="71"/&gt;     &lt;mif:graphEdge&gt;       &lt;mif:anchor         connectToShapeId="Sheet.189"&gt;         &lt;mif:position           x="785" y="137"/&gt;         &lt;/mif:anchor&gt;       &lt;/mif:graphEdge&gt;     &lt;/mif:anchorage&gt;   &lt;/mif:graphElement&gt; &lt;/entryPoint&gt;</code>	Entry-Point
<code>&lt;class className="A_CoverageUniversal"   presentation="HL7"&gt; &lt;mif:graphElement   containerDiagramName="Main"   isHeightAutoSize="true"   isWidthAutoSize="true"   shapeId="Sheet.93"   shapeTemplate="CMET"&gt;   &lt;mif:position x="338" y="204"/&gt;   &lt;mif:size height="60" width="105"/&gt; &lt;/mif:graphElement&gt; &lt;/class&gt;</code>	Klassen

Ausschnitt aus PRPA_MT201301UV.mif	Kommentar
...	weitere Klassen
<pre> &lt;association   className="E_PlaceInformational"   relationshipName="birthplace"&gt; </pre>	Beginn Assoziation
<pre> &lt;mif:graphElement   connectToShapeId="Sheet.447"   containerDiagramName="Main"&gt;   &lt;mif:position x="1273" y="918"/&gt;   &lt;mif:graphEdge&gt;     &lt;mif:anchor       connectToShapeId="Sheet.7"&gt;       &lt;mif:position x="1197" y="1001"/&gt;     &lt;/mif:anchor&gt;   &lt;/mif:graphEdge&gt; &lt;/mif:graphElement&gt; </pre>	
<pre> &lt;/association&gt; </pre>	
...	weitere Assoziationen
<pre> &lt;/mif:graphicRepresentation&gt; </pre>	
<pre> &lt;derivedFrom staticModelDerivationId="1"&gt;   &lt;mif:targetStaticModel artifact="RIM"     root="DEFN" version="00"/&gt; &lt;/derivedFrom&gt; </pre>	derivedFrom
.. <derivedFrom/> ..	weitere derivedFrom
<pre> &lt;containedClass&gt;   &lt;mif:commonModelElementRef     name="A_CoverageUniversal"&gt;     &lt;derivedFrom       className="FinancialContract"       staticModelDerivationId="1"/&gt;     &lt;derivedFrom       className="A_CoverageUniversal"       staticModelDerivationId="2"/&gt;     &lt;derivedFrom       className="A_CoverageUniversal"       staticModelDerivationId="3"/&gt;     &lt;supplierVocabulary&gt;       &lt;vocabularyDomain xmlns=""         name="ActClass"/&gt;     &lt;/supplierVocabulary&gt;     &lt;cmet name="COCT_MT180000UV04"/&gt;   &lt;/mif:commonModelElementRef&gt; &lt;/containedClass&gt; </pre>	contained classes (Beispiel 1)

Ausschnitt aus PRPA_MT201301UV.mif	Kommentar
<pre> &lt;containedClass&gt;   &lt;mif:class isAbstract="false"     name="AdministrativeObservation"&gt;     &lt;derivedFrom className="Observation"       staticModelDerivationId="1"/&gt;     &lt;mif:attribute conformance="R"       isImmutable="true"       isMandatory="true"       maximumMultiplicity="1"       minimumMultiplicity="1"       name="classCode"       sortKey="1"&gt;       &lt;derivedFrom         attributeName="classCode"         className="Observation"         staticModelDerivationId="1"/&gt;       &lt;derivedFrom         attributeName="classCode"         className="AdministrativeObservation"         staticModelDerivationId="2"/&gt;       &lt;derivedFrom         attributeName="classCode"         className="AdministrativeObservation"         staticModelDerivationId="3"/&gt;       &lt;mif:type name="CS"/&gt;       &lt;supplierVocabulary         codingStrength="CNE"&gt;         &lt;code code="OBS"           codeSystemName="ActClass"/&gt;       &lt;/supplierVocabulary&gt;     &lt;/mif:attribute&gt;     &lt;mif:attribute conformance="R"       isImmutable="true"       isMandatory="true"       maximumMultiplicity="1"       minimumMultiplicity="1"       name="moodCode" sortKey="2"&gt;       &lt;enumerationValues&gt;EVN&lt;/enumerationValues&gt;       &lt;derivedFrom attributeName="moodCode"         className="Observation"         staticModelDerivationId="1"/&gt;       &lt;derivedFrom attributeName="moodCode"         className="AdministrativeObservation" </pre>	<p>(Beispiel 2)</p> <p>mit Attributen, die auszulesen sind.</p> <p>Attributname</p> <p>Datentyp des Attributs</p> <p>Vokabeldomäne</p>

Ausschnitt aus PRPA_MT201301UV.mif	Kommentar
<pre> staticModelDerivationId="2"/&gt; &lt;derivedFrom attributeName="moodCode" className="AdministrativeObservation" staticModelDerivationId="3"/&gt; &lt;mif:type name="CS"/&gt; &lt;supplierVocabulary codingStrength="CNE"&gt; &lt;code code="EVN" codeSystemName="ActMood"/&gt; &lt;/supplierVocabulary&gt; &lt;/mif:attribute&gt; &lt;mif:attribute isImmutable="false" isMandatory="false" maximumMultiplicity="*" minimumMultiplicity="0" name="id" sortKey="3"&gt; &lt;derivedFrom attributeName="id" className="Observation" staticModelDerivationId="1"/&gt; &lt;derivedFrom attributeName="id" className="AdministrativeObservation" staticModelDerivationId="2"/&gt; &lt;derivedFrom attributeName="id" className="AdministrativeObservation" staticModelDerivationId="3"/&gt; &lt;mif:type name="SET"&gt; &lt;argumentDatatype name="II"/&gt; &lt;/mif:type&gt; &lt;/mif:attribute&gt; &lt;mif:attribute conformance="R" isImmutable="false" isMandatory="true" maximumMultiplicity="1" minimumMultiplicity="1" name="code" sortKey="4"&gt; &lt;derivedFrom attributeName="code" className="Observation" staticModelDerivationId="1"/&gt; &lt;derivedFrom attributeName="code" className="AdministrativeObservation" staticModelDerivationId="2"/&gt; &lt;derivedFrom attributeName="code" </pre>	<p>zusammengesetzter Datentyp</p>

Ausschnitt aus PRPA_MT201301UV.mif	Kommentar
<pre>         className="AdministrativeObservation"         staticModelDerivationId="3"/&gt;     &lt;mif:type name="CD"/&gt;     &lt;supplierVocabulary         codingStrength="CWE"&gt;         &lt;vocabularyDomain             name="AdministrativeObservationType"/&gt;         &lt;/supplierVocabulary&gt;     &lt;/mif:attribute&gt;     &lt;mif:attribute isImmutable="false"         isMandatory="false"         maximumMultiplicity="1"         minimumMultiplicity="0"         name="statusCode" sortKey="5"&gt;     &lt;derivedFrom         attributeName="statusCode"         className="Observation"         staticModelDerivationId="1"/&gt;     &lt;derivedFrom         attributeName="statusCode"         className="AdministrativeObservation"         staticModelDerivationId="2"/&gt;     &lt;derivedFrom         attributeName="statusCode"         className="AdministrativeObservation"         staticModelDerivationId="3"/&gt;     &lt;mif:type name="CS"/&gt;     &lt;supplierVocabulary         codingStrength="CNE"&gt;         &lt;vocabularyDomain             name="ActStatus"/&gt;         &lt;/supplierVocabulary&gt;     &lt;/mif:attribute&gt;     &lt;mif:attribute isImmutable="false"         isMandatory="false"         maximumMultiplicity="1"         minimumMultiplicity="0"         name="effectiveTime"         sortKey="6"&gt;     &lt;derivedFrom         attributeName="effectiveTime"         className="Observation"         staticModelDerivationId="1"/&gt; </pre>	

	Ausschnitt aus PRPA_MT201301UV.mif	Kommentar
	<pre> &lt;derivedFrom     attributeName="effectiveTime"     className="AdministrativeObservation"     staticModelDerivationId="2"/&gt; &lt;derivedFrom     attributeName="effectiveTime"     className="AdministrativeObservation"     staticModelDerivationId="3"/&gt; &lt;mif:type name="GTS"/&gt; &lt;/mif:attribute&gt; &lt;mif:attribute conformance="R"     isImmutable="false"     isMandatory="true"     maximumMultiplicity="1"     minimumMultiplicity="1"     name="value"     sortKey="7"&gt; &lt;derivedFrom attributeName="value"     className="Observation"     staticModelDerivationId="1"/&gt; &lt;derivedFrom attributeName="value"     className="AdministrativeObservation"     staticModelDerivationId="2"/&gt; &lt;derivedFrom attributeName="value"     className="AdministrativeObservation"     staticModelDerivationId="3"/&gt; &lt;mif:type name="ANY"/&gt; &lt;supplierVocabulary     codingStrength="CWE"&gt;     &lt;vocabularyDomain         name="ObservationValue"/&gt;     &lt;/supplierVocabulary&gt; &lt;/mif:attribute&gt; &lt;/mif:class&gt; &lt;/containedClass&gt; </pre>	
...		weitere contained classes
	<pre> &lt;association sortKey="5"&gt;     &lt;mif:connections&gt;         &lt;mif:traversableConnection             conformance="R"             isMandatory="false"             maximumMultiplicity="1" </pre>	

Ausschnitt aus PRPA_MT201301UV.mif	Kommentar
<pre> minimumMultiplicity="1" name="careGiverPerson" participantClassName=   "E_PersonInformational" sortKey="AZ"&gt; &lt;derivedFrom   associationEndName="player"   className="Role"   staticModelDerivationId="1"/&gt; &lt;derivedFrom   associationEndName="careGiverPerson"   className="CareGiver"   staticModelDerivationId="2"/&gt; &lt;derivedFrom   associationEndName="careGiverPerson"   className="CareGiver"   staticModelDerivationId="3"/&gt; &lt;/mif:traversableConnection&gt; &lt;mif:nonTraversableConnection   participantClassName="CareGiver"&gt;   &lt;derivedFrom     associationEndName="playedRole"     className="Entity"     staticModelDerivationId="1"/&gt;   &lt;/mif:nonTraversableConnection&gt; &lt;/mif:connections&gt; &lt;/association&gt; </pre>	
...	weitere Assoziationen
</mif:staticModel>	

### 21.1.3.2. Ontologie-Struktur

Tabelle 34: Ontologie-Struktur für R-MIMs

MessageElement	
RMIM-Part	Oberklasse für Details der RMIMs
RMIM-Class	
<RMIM>	dedizierte RMIMs
<RMIM>.<Klasse>	Klassen aus den RMIMs.
RMIM-Attribute	Attribute zu den RMIMs: separate Hierarchie mit Querverweisen
<RMIM>-Attr	



<RMIM>.<Attribut>
-------------------

### 21.1.3.3. importierte Ontologien

Folgende Ontologien (OWL-Dateien) werden importiert:

- Vocabulary
  - codeSystem
  - conceptDomain
  - valueSet

### 21.1.3.4. Integritätsregeln

Da hier auf alle Vokabeldomänen verwiesen werden kann, müssen diese bekannt sein. Deshalb müssen die Vokabeldomänen insofern komplett aufgelistet und importiert werden. Auf die dazugehörigen Details kann dann aber verzichtet werden.

### 21.1.4. Datentypen

Die Datentypen werden derzeit nur im Release 1.0 bereitgestellt.

#### 21.1.4.1. MIF-Datei

Im Folgenden wird die MIF-Struktur für Datentypen wiedergegeben.

Tabelle 35: MIF-Struktur für Datentypen

Datentypen	Kommentar
<?xml version="1.0" encoding="UTF-8"?>	XML-Header
<pre>&lt;mif:datatypeModelLibrary   xmlns:xsi=<a href="http://www.w3.org/2001/XMLSchema-instance">http://www.w3.org/2001/XMLSchema-instance</a>   xsi:schemaLocation=     "urn:hl7-org:v3/mif2 ../../../../mif-COMplete.xsd"   xmlns:mif="urn:hl7-org:v3/mif2"   xmlns:html=<a href="http://www.w3.org/1999/xhtml">http://www.w3.org/1999/xhtml</a>   packageKind="version"   name="1.0"   title="Data Types - Abstract Specification"   schemaVersion="2.1.1"&gt;</pre>	
<pre>&lt;mif:packageLocation combinedId="DEFN=UV=DT=1.0"   root="DEFN" artifact="DT" realmNamespace="UV"   version="1.0"/&gt;</pre>	
<mif:header>	Header
<pre>&lt;mif:contributor&gt;   &lt;mif:role&gt;Chair/Editor&lt;/mif:role&gt;   &lt;mif:name name="Gunther Schadow"/&gt;   &lt;mif:affiliation&gt;</pre>	Beitragende

Datentypen	Kommentar
<pre>                 Regenstrief Institute             &lt;/mif:affiliation&gt;             &lt;mif:email&gt;....&lt;/mif:email&gt;             &lt;/mif:contributor&gt;         </pre>	
...	
<pre>             &lt;mif:approvalInfo                 approvalStatus="Approved Normative Standard"                 ballotOccurrence="2"/&gt;         </pre>	
<pre>         &lt;/mif:header&gt;     </pre>	
<pre>             &lt;mif:importedVocabularyModelPackage                 combinedId="DEFN=UV=VO"                 root="DEFN" realmNamespace="UV"                 artifact="VO"/&gt;         </pre>	
<pre>             &lt;mif:datatype                 name="PN" title="PersonName"                 datatypeKind="Definition"                 visibility="public"&gt;         </pre>	Datentyp
<pre>             &lt;mif:derivedFrom relationship="restriction"&gt;                 &lt;mif:targetDatatype name="EN"/&gt;             &lt;/mif:derivedFrom&gt;         </pre>	Spezialisierungshierarchie
<pre>             &lt;mif:annotations&gt;                 &lt;mif:documentation&gt;                     &lt;mif:definition&gt;                         &lt;mif:text&gt;                             ...                         &lt;/mif:text&gt;                     &lt;/mif:definition&gt;                     &lt;mif:designComments&gt;                         &lt;mif:text&gt;                             ....                         &lt;/mif:text&gt;                     &lt;/mif:designComments&gt;                     &lt;mif:otherAnnotation type="DTDLD"&gt;                         &lt;mif:data&gt;                             type PersonName alias PN specializes EN;                         &lt;/mif:data&gt;                     &lt;/mif:otherAnnotation&gt;                 &lt;/mif:documentation&gt;             &lt;/mif:annotations&gt;         </pre>	Annotation
<pre>         &lt;/mif:datatype&gt;     </pre>	

Datentypen	Kommentar
...	Weitere Datentypen
</mif:datatypeModelLibrary>	

### 21.1.4.2. Ontologie-Struktur

Tabelle 36: Ontologie-Struktur für R-MIMs

MessageElement	
DataStructure	Oberklasse für Details der Datentypen
DataType	
DataTypeAttribute	Attribute zu den Datentypen
DataTypeParameter	Parameter für die Attribute

### 21.1.5. CMET

Die Zusammenstellung aller definierten CMETs ist in einer speziellen Datei „cmetList.mif“ aufgeführt. Die dazugehörigen Details sind wiederum R-MIMs, die dann in der entsprechenden R-MIM-Spezifikation zu finden sind.

#### 21.1.5.1. MIF-Datei

Im Folgenden wird die MIF-Struktur für CMETs wiedergegeben.

Tabelle 37: MIF-Struktur für CMETs

CMET	Kommentar
<?xml version="1.0" encoding="UTF-8"?>	XML-Header
<mif:staticModelInterfacePackage xmlns:mif="urn:hl7-org:v3/mif" name="CMETinfo" packageKind="name" schemaVersion="2.0">	Header
<mif:commonModelElementDefinition name="E_PersonUniversal" attributionLevel="universal" entryKind="Entity">	Einzelnes CMET
<mif:supplierVocabulary> <mif:code code="PSN" codeSystem="2.16.840.1.113883.5.41"/> </mif:supplierVocabulary>	
<mif:annotations> ....	Anmerkungen

CMET	Kommentar
<code>&lt;/mif:annotations&gt;</code>	
<code>&lt;mif:boundStaticModel   root="DEFN" section="IM"   subSection="CO" domain="CT"   artifact="MT-deprecated"   realmNamespace="UV"   id="030200"/&gt;</code>	
<code>&lt;mif:entryClass name="Person"/&gt;</code>	Entry-Point
<code>&lt;/mif:commonModelElementDefinition&gt;</code>	
...	Weitere CMETs
<code>&lt;/mif:staticModelInterfacePackage&gt;</code>	

### 21.1.5.2. Ontologie-Struktur

Tabelle 38: Ontologie-Struktur für CMETs

MessageElement	
CMET-Part	
CMET	

## 21.2. Umsetzung der verschiedenen HL7-v2-Strukturen nach OWL

Die Tabellenattribute aus der HL7-Datenbank werden gemäß nachfolgender Mapping-tabelle umgesetzt:

Mappingtabelle für die Umsetzung der Attribute in eine Ontologie:

Tabelle 39: Umsetzung der HL7-v2-Stukturen nach OWL

ER-Modell	OWL/Ontologie	Wie
description	<code>rdfs:comment xml:lang="en"</code>	als Annotation zu einer Klasse
interpretation	<code>rdfs:comment xml:lang="de"</code>	als Annotation zu einer Klasse
req_opt	BasicConcept -> Optionality BasicConcept -> Condition <code>owl:minCardinality</code>	als Klasse (Basis-Konzept in CSO) sowie als Property
repetitions	<code>owl:maxCardinality</code>	als Property
length	<code>DatatypeProperty</code>	als Property
table	<code>owl:Class</code>	als Klasse
table_value	<code>owl:Class</code>	als Klasse
data_item	<code>owl:Class</code>	als Klasse
seg_code	<code>owl:Class</code>	als Klasse
data_structure	<code>owl:Class</code>	als Klasse

event_code	owl:Class	als Klasse
Section	rdfs:comment xml:lang="sec"	als Annotation

## 22. Verzeichnisse

### 22.1. Abkürzungen

In der Arbeit wird eine Reihe von Abkürzungen verwendet, die kurz erläutert werden sollen:

Tabelle 40: Abkürzungsverzeichnis

Abkürzung:	Bedeutung:
ACGT	Advancing Clinico Genomic Trials on Cancer
ADT	Abrechnungsdatenträger (s. xDT)
ADT	Patientenstammdaten (Admission, Discharge and Transfer)
ANSI	American National Standards Institute
API	Application Programming Interface
ARB	Architecture Review Board
ASCII	American Standard Code for Information Interchange
ASN.1	Abstract Syntax Notation One
ASTM	American Society for Testing and Materials
BDT	Behandlungsdatenträger (s. xDT)
BFO	Basic Formal Ontology
BG	Berufsgenossenschaft
CCOW	Clinical Context Object Working Group
CDA	Clinical Document Architecture
CMET	Common Message Element Types
CNE	Coded With No Exceptions (Datentyp in V2.x.; Coding Strength in V3)
CSO	Communication Standards Ontology
CORBA	Common Object Request Broker Architecture
CWE	Coded With Exceptions (Datentyp in V2.x.; Coding Strength in V3)
DALE-UV	Datenträgeraustausch Leistungserbringer in der Unfallversorgung
D-MIM	Domain Message Information Model
D2D	Doctor-to-Doctor Communication
DICOM	Digital Imaging and Communication in Medicine
DIN	Deutsches Institut für Normung e.V.
DSTU	Draft Standard for Trial Use
EBCDIC	Extended Binary Coded Decimal Interchange Code
ebXML	e-business eXtended Markup Language
EDI	Electronic Data Interchange

Abkürzung:	Bedeutung:
EDIFACT	Electronic Data Interchange for Administration, Commerce and Transport
ER7	Encoding Rules 7 (Standard-Datenformat für HL7 Version 2.x)
GCM	Generic Component Model
GCS	Glasgow Coma Scale
GDT	Gerätedatenträger (s. xDT)
HBA	Heilberufsausweis (=HPC)
HCM	Hospital Communication Module
HDF	HL7 Development Framework
HL7	Health Level Seven
HMD	Hierarchic Message Definition
HPC	Health Professional Card
http	Hypertext Transfer Protocol
H.PR.I.M.	Harmoniser et PRo mouvoir les Informatique Médicales [H.PR.I.M] (Konkurrenzkommunikationstandard zu HL7 in Frankreich)
ICD	International Classification of Diseases
IHE	Integrating the Healthcare Enterprise [IHE]
InM	Infrastructure and Messaging (Arbeitsgruppe bei HL7)
ISO	International Organisation for Standardisation
ITS	Implementation Technology Specification
KIS	Krankenhausinformationssystem
KVDT	Kassenärztliche Vereinigung Datentransfer
MDA	Model Driven Architecture
MDF	Messaging Developing Framework
MSH	Message Header (Nachrichtenkopf in HL7 v2.x)
N3	Notation 3
NIST	National Institute for Standardisation ( <a href="http://www.nist.gov">http://www.nist.gov</a> )
OASIS	Organization for the Advancement of Structured Information Standards
ODA	Ontology Driven Architecture
OID	Object Identifier
OMG	Object Management Group
OPPL	Ontology Preprocessor Language
OSGi	Open Services Gateway Initiative
OSI	Open Standards Interconnection
OWL	Web Ontology Language
PEO	Profilier Enforcement Organisation
R-MIM	Refined Message Information Model
RDF	Resource Description Framework
RDFS	RDF Schema

Abkürzung:	Bedeutung:
RIM	Referenz Information Model
RM-ODP	Reference Model for Open Distributed Processing
RO	Relationship Ontology
SAEAF	Services Aware Enterprise Architecture Framework
SAIF	Service Architecture Interoperability Framework
SCIPHOX	Standardized Communication In Physician Offices and HOspitals using XML
SDO	Standard Developing Organisation
SMD	Static Model Designer
SNOMED	Systemized NOmenclature in MEDicine
SNOMED CT	Snomed Clinical Terms
SOAP	Simple Object Access Protocol
SPARQL	Abfragesprache für RDF
SWRL	Semantic Web Rule Language
UML	Unified Modeling Language
UN/EDIFACT	United Nations Electronic Data Interchange for Administration, Commerce and Transport
UNICODE	Universeller Zeichensatz [Unicode]
UTF	Universal Multiple-Octet Coded Character Set (UCS) Transformation Format
VB	Visual Basic
VBA	Visual Basic for Applications
VCS	VdAP Communication Standard
W3C	World Wide Web Consortium
WHO	World Health Organisation
xDT	Datenträgeraustausch (als Familie von Spezifikationen)
XML	eXtensible Markup Language
XSL	XML Stylesheet Language
XSLT	XSL Translation

## **22.2. Abbildungen**

<b>Abbildungsverzeichnis.....</b>	<b>Seite</b>
Abbildung 1: Aktuelle Kommunikationsszenarien im Gesundheitswesen.....	9
Abbildung 2: Die 4 Seiten einer Nachricht nach Schulz von Thun [SchuVT] .....	10
Abbildung 3: Informationszyklus [Blob2007c] .....	11
Abbildung 4: Ereignisorientierte Kommunikation.....	15
Abbildung 5: Ebenenorientierte Auflistung angewandter Kommunikationsstandards .....	15
Abbildung 6: Datenaustausch mit HL7.....	18

<b>Abbildungsverzeichnis.....</b>	<b>Seite</b>
Abbildung 7: Zusammenhang mit Zeichen (nach [Morris1938]).....	23
Abbildung 8: semiotisches Dreieck [Ogde1923].....	24
Abbildung 9: Repräsentationsformen für Ontologien [BloOem2009] .....	26
Abbildung 10: Wissenspyramide (nach [AamNyg1995]) .....	27
Abbildung 11: Die Sichten des RM-ODP .....	33
Abbildung 12: "Incremental Approach to Working Interoperability" .....	35
Abbildung 13: Das generische Komponentenmodell [Blob2002, Blob2007b].....	36
Abbildung 14: Einschränkung des generischen Komponentenmodell [Blob2002, Blob2007b] .....	39
Abbildung 15: Grundlagen der Fachdomänen .....	40
Abbildung 16: Reduktion des GCM auf den Enterprise View [Blob2002, Blob2007b] .....	41
Abbildung 17: Ontologie-Typen [Guar1997, Guar1998] .....	44
Abbildung 18: Ontologie-Typen [Blob2009b].....	45
Abbildung 19: Mikas Pyramide der zunehmenden Formalisierung [MiAk2004] .....	46
Abbildung 20: UML-Klassendiagramm der Begriffsmengen.....	54
Abbildung 21: Spezialisierungen der assoziativen Relation [Blob1987] .....	55
Abbildung 22: Einordnung in das GCM (1) .....	56
Abbildung 23: Einordnung in das GCM (2) .....	56
Abbildung 24: UML-HL7 v2.x Modell.....	60
Abbildung 25: Vereinfachte schematische Zuordnung der Nachrichtenstruktur .....	61
Abbildung 26: UML-HL7 v2.x Datentypen Modell .....	63
Abbildung 27: UML-HL7 V3 Modell .....	69
Abbildung 28: Angepasste Ontologie-Typen .....	70
Abbildung 29: Snomed CT – Inhaltsrepräsentation mittels dreier Tabellen .....	72
Abbildung 30: Aufteilung der Ontologielevels (für die hier geforderte Architektur) .....	74
Abbildung 31: Ontologie-Struktur .....	75
Abbildung 32: Information Artefact Ontology mit Relationen (Darstellung in Protégé) .....	76
Abbildung 33: Basis-Ontologie: Konzepthierarchie (1. Entwurf als Darstellung in Protégé) .....	77
Abbildung 34: RO: ObjectProperties.....	78
Abbildung 35: Vergleich mit anderen Arbeiten.....	79
Abbildung 36: Erweiterte Top-Level Ontologie (Darstellung in Protégé) .....	81
Abbildung 37: CSO (Darstellung in Protégé).....	83
Abbildung 38: Basis-Konzepte von CSO (Darstellung in Protégé) .....	84
Abbildung 39: Optionalität (Darstellung in Protégé) .....	85
Abbildung 40: Relationen (Darstellung in Protégé).....	86
Abbildung 41: HL7-V3-Tooling-Sicht (nach McKenzie) .....	88



<b>Abbildungsverzeichnis.....</b>	<b>Seite</b>
Abbildung 42: Darstellung von HL7 V3 im GCM .....	89
Abbildung 43: HL7-V3-Elemente in OWL (dargestellt in Protégé) .....	95
Abbildung 44: Ableitungsprozess der geforderten OWL-Spezifikationen für HL7 v2.x.....	97
Abbildung 45: Einsortierung von HL7 V2.x in das GCM .....	98
Abbildung 46: Klassenhierarchieübersicht für HL7 v2.6 in Protégé (OWL) .....	100
Abbildung 47: Relationshierarchie für HL7 v2.6 in Protégé (OWL) .....	101
Abbildung 48: Detailklassenhierarchie für HL7 v2.6 in Protégé (OWL) .....	104
Abbildung 49: Ontologie für HL7 v2.x (UML) .....	106
Abbildung 50: Ontologie für HL7 v2.x: PID-8 (Protégé) .....	107
Abbildung 51: Konsistenzregel in OWL und Protégé .....	109
Abbildung 52: Struktur in der Fachdomäne .....	109
Abbildung 53: Name, Adresse und Telefonnummer .....	111
Abbildung 54: Ontologische Konzepte für Namen, Adressen und Telefonnummern .....	112
Abbildung 55: UML-Modell für Fall (als Extrakt) .....	114
Abbildung 56: Diagnose .....	115
Abbildung 57: HL7 V3 Modell für Scores .....	116
Abbildung 58: UML-Modell für Scores .....	117
Abbildung 59: Encounter and Registration in ACGT-FO .....	125
Abbildung 60: UML OWL Modell .....	126
Abbildung 61: Protégé OWL Modell (als direkte Umsetzung) .....	126
Abbildung 62: Protégé OWL Modell für Scores (in Bezug auf ACGT) .....	127
Abbildung 63: Code-Definition für die Scores auf Basis von ACGT .....	128
Abbildung 64: Reference Ranges für Scores auf Basis von ACGT.....	128
Abbildung 65: Einschränkungen für Scores in Protégé .....	129
Abbildung 66: Beispiel-Mapping V3-v2.x (als direktes Mapping) .....	131
Abbildung 67: Beispiel-Mapping V2.x-ACGT-V3 (indirektes Mapping) .....	131
Abbildung 68: Beispiel-Mapping V3-v2.x (mit Zerlegung) .....	132
Abbildung 69: Mapping .....	133
Abbildung 70: Komponentenorientiertes Mapping .....	134
Abbildung 71: Basis-Ontologie: Mapping (Darstellung in Protégé) .....	136
Abbildung 72: Basis-Ontologie: Relationen zum Mapping (Darstellung in Protégé) .....	136
Abbildung 73: Funktionsweise der Regular Expressions .....	137
Abbildung 74: Regular Expressions (Darstellung in Protégé) .....	137
Abbildung 75: Mapping auf Regular Expressions (Darstellung in Protégé) .....	139
Abbildung 76: graphische Darstellung der Regular Expressions (Darstellung in Protégé) .....	139
Abbildung 77: Data-Properties (Darstellung in Protégé).....	140
Abbildung 78: v2-Mapping (Darstellung in Protégé) .....	141

<b>Abbildungsverzeichnis.....</b>	<b>Seite</b>
Abbildung 79: v2-Mapping (Darstellung in Protégé) .....	141
Abbildung 80: v2-Mapping mit Bedingungen (Darstellung in Protégé).....	142
Abbildung 81: v2-Mapping mit Bedingungen (Darstellung mit OwlPropViz).....	142
Abbildung 82: Übersichtsschaubild (Darstellung in Protégé mit OwlPropViz).....	143
Abbildung 83: V3-Mapping (Darstellung in Protégé) .....	145
Abbildung 84: Erzeugung der Ontologien .....	146
Abbildung 85: Ontologie-Import-Struktur .....	147
Abbildung 86: Formular zur Steuerung der HL7-v2.x-OWL-Generierung .....	152
Abbildung 87: Formular zur Steuerung der HL7-V3-OWL-Generierung .....	153
Abbildung 88: UML-Business-Objekte.....	168
Abbildung 89: UML-Datenbankmodell.....	169
Abbildung 90: Umsetzungsprozess.....	170
Abbildung 91: Architekturskizze .....	172
Abbildung 92: Beispiel für einfache semantische Aussagen .....	175
Abbildung 93: OWL-Pizza-Beispiel in Protégé .....	178

## **22.3. Tabellen**

<b>Tabellenverzeichnis .....</b>	<b>Seite</b>
Tabelle 1: Gegenüberstellung der verschiedenen Standards .....	16
Tabelle 2: ISO/OSI-Ebenen [DIN 7498].....	18
Tabelle 3: Interoperability Levels [Blob2007b] .....	19
Tabelle 4: Klassifikation: Alpha-IDs und ICD-Codes .....	30
Tabelle 5: Domänenspezifische Beobachtungseinheiten [Blob1987] .....	50
Tabelle 6: Präsentation existierender Ansätze .....	79
Tabelle 7: CSO + BFO .....	82
Tabelle 8: Umsetzungsmöglichkeiten für V3 je nach Datenquelle .....	87
Tabelle 9: Fehler in der MIF-Definition des RIM 0219 .....	96
Tabelle 10: Fehler in der MIF-Definition des R-MIM .....	96
Tabelle 11: Informationselemente von HL7 v2.x und deren Umsetzung in OWL .....	99
Tabelle 12: BFO Modell/Hierarchie .....	118
Tabelle 13: ACGT + BFO Modell/Hierarchie (generically_dependent_continuant).....	119
Tabelle 14: ACGT + BFO Modell/Hierarchie (quality) .....	120
Tabelle 15: ACGT + BFO Modell/Hierarchie (role) .....	121
Tabelle 16: ACGT + BFO Modell/Hierarchie (independent_continuant).....	122

---

<b>Tabellenverzeichnis .....</b>	<b>Seite</b>
Tabelle 17: Addendum zu CSO .....	135
Tabelle 18: Ontologiegrundstruktur für die Kommunikationsstandards .....	147
Tabelle 19: Ontologiegrundstruktur für das Mapping .....	148
Tabelle 20: Vergleich und Bewertung der Umsetzungsansätze.....	149
Tabelle 21: Eine kleine Statistik der Klassen und Beziehungen .....	159
Tabelle 22: Dateiliste der entwickelten Ontologien .....	160
Tabelle 23: Dateiliste der benutzten BFO-Ontologien.....	161
Tabelle 24: Bewertung der eigenen Arbeit (vorher/nachher) .....	165
Tabelle 25: Bewertung der eigenen Arbeit (Vor-/Nachteile).....	166
Tabelle 26: RDF-Statements .....	175
Tabelle 27: OWL1 Sprachkonstrukte .....	178
Tabelle 28: OWL DL Sprachkonstrukte .....	179
Tabelle 29: MIF-Struktur für Elemente des RIMs.....	181
Tabelle 30: Ontologie-Struktur für das RIM .....	184
Tabelle 31: MIF-Struktur für Vocabulary .....	185
Tabelle 32: Ontologie-Struktur für Vocabulary .....	189
Tabelle 33: MIF-Struktur für R-MIM .....	189
Tabelle 34: Ontologie-Struktur für R-MIMs.....	196
Tabelle 35: MIF-Struktur für Datentypen .....	197
Tabelle 36: Ontologie-Struktur für R-MIMs.....	199
Tabelle 37: MIF-Struktur für CMETs .....	199
Tabelle 38: Ontologie-Struktur für CMETs.....	200
Tabelle 39: Umsetzung der HL7-v2-Stukturen nach OWL.....	200
Tabelle 40: Abkürzungsverzeichnis .....	201

## 22.4. Literatur

- [§301] §301 SGB V: "Sozialgesetzbuch Fünftes Buch, Gesetzliche Krankenversicherung: §301 Krankenhäuser"  
[http://www.sozialgesetzbuch.de/gesetze/05/index.php?norm\\_ID=0530100](http://www.sozialgesetzbuch.de/gesetze/05/index.php?norm_ID=0530100), zuletzt zugegriffen am 5.5.08
- [AamNyg1995] Aamodt A, Nygard M: "Different roles and mutual dependencies of data, information and knowledge"  
in "Data and Knowledge Engineering 16, S.191-222, 1995
- [ACGT] "Advancing Clinico Genomic Trials on Cancer"  
Project ID: FP6-IST-026996; European Commission, 02/06-01/10  
<http://www.eu-acgt.org/>  
<http://www.ifomis.org/wiki/ACGT>
- [ACR-NEMA 2003] ACR-NEMA. DICOM. 2003  
<http://medical.nema.org/> und <http://www.rsna.org/>
- [AHML] AHML: Australian Healthcare Messaging Laboratory  
<http://www.ahml.com.au>
- [AllHen2008] Allemang D, Hendler J: "Semantic Web For the Working Ontologist in RDF, RDFS and OWL", Morgan Kaufmann, 2008, ISBN 978-0-12-373556-0
- [ANSI] ANSI: American National Standards Institute  
<http://www.ansi.org>
- [Arms1973] Armstrong DM: "Belief, Truth and Knowledge", Cambridge: Cambridge University Press, 1973
- [Artemis] Artemis-Project: "A Semantic Web Service-based P2P Infrastructure for the interoperability of Medical Information Systems",  
<http://www.srdc.metu.edu.tr/webpage/projects/artemis/>
- [ASCII] "American Standard Code for Information Interchange"  
<http://www.asphelper.de/referenz/ASCIANSI.asp>
- [ASC X12] Accredited Standards Committee X12  
<http://www.x12.org/> (zuletzt zugegriffen am 5.5.08)  
<http://www.disa.org/x12org/about/index.cfm> (zuletzt zugegriffen am 5.5.08)
- [ASN.1] Abstract Syntax Notation One  
<http://www.asn1.org>
- [ASTM] ASTM International (American Society for Testing and Materials)  
<http://www.astm.org>
- [BaaCal2007] Baader, Calvanese, McGuinness, Nardi und Patel-Schneider (Eds): "The Description Logic Handbook", 2nd Edition, Cambridge University Press, 2007
- [Bern2001] Berners-Lee T: "The Semantic Web":  
Scientific American, May 2001  
<http://www.sciam.com/article.cfm?id=the-semantic-web&print=true>,  
zuletzt zugegriffen 13.03.09  
"Web Future": <http://www.w3.org/2006/Talks/0314-ox-tbl/>, zuletzt zugegriffen 13.03.09
- [BFO] BFO: "Basic Formal Ontology"  
<http://www.ifomis.org/bfo>

- 
- [Bice2005] Bicer et al.: "Providing Semantic Interoperability in the Healthcare Domain through Ontology Mapping", 2005, IOS-Press
- [Blob1987] Blobel: Vorlesung Medizinische Informatik, Universität Magdeburg, 1987
- [Blob1997] Blobel B: "Assessment of Middleware Concepts Using a Generic Component Model"  
Proceedings of the Conference "Toward An Electronic Health Record Europe '97", 1997, pp. 221-228. London
- [Blob2000] Blobel B: "Application of the Component Paradigm for Analysis and Design of Advanced Health System Srchitectures"  
International Journal of Medical Informatics 2000, 60, 3: pp.281-301
- [Blob2002] Blobel B: "Analysis, Design and Implementation for Secure and Interoperable Distributed Health Information Systems"  
Series Studies in Health Technology and Informatics, Amsterdam: Vol.89, IOS Press, 2002, ISBN: 1 58603 277
- [Blob2003] Blobel B, Pharow P (Edrs.): "Advanced Health Telematics and Telemedicine."  
The Magdeburg Expert Summit Textbook.  
Series "Studies in Health Technology and Informatics" Vol. 96.  
Amsterdam: IOS Press, 2003
- [Blob2005] Blobel B, Pharow P: "A Model-Driven Approach for the German Health Telematics Architectural Framework and the Related Security Infrastructure" In: Engelbrecht R, Geissbuhler A, Lovis Ch, Mihalas G (Edrs.): Connecting Medical Informatics and Bio-Informatics.  
Proceedings of MIE 2005, pp. 391-396. Series Studies in Health Technology and Informatics, Vol. 116, IOS Press, Amsterdam, Berlin, Oxford, Tokyo, Washington DC, ISBN: 1-58603-549-5  
extendedly published in International Journal of Medical Informatics 76, 2-3 (2007) pp. 169-175
- [Blob2006] Blobel B: "Concepts Representation in Health Informatics for Enabling Intelligent Architectures" In: Hasman A, Haux R, van der Lei J, De Clercq E, Roger-France F (Edrs.): Ubiquity: Technology for Better Health in Aging Societies, pp. 285-291. Series Studies in Health Technology and Informatics, Vol. 124. IOS Press, Amsterdam, ISBN: 1-58603-549-5
- [BloEng2006] Blobel B, Engel K, Pharow P: "Semantic Interoperability – HL7 Version 3 Compared to Advanced Architecture Standards"  
Methods Inf Med 2006; 45: pp 343-353]
- [Blob2007a] Blobel B: "Comparing approaches for advanced security infrastructures",  
International Journal of Medical Informatics 76 (2007), S.454-459
- [Blob2007b] Blobel B: "Introduction into Advanced eHealth – The Personal Health Challenge", In: Blobel B, Pharow P, Nerlich M (Edrs.): eHealth: Combining Health Telematics, Telemedicine, Biomedical Engineering and Bioinformatics to the Edge - Global Experts Summit Textbook, pp. 3-14. Series "Studies in Health Technology and Informatics", Vol. 134. IOS Press, Amsterdam, Berlin, New York, Tokyo. ISBN-978-1-58603-835-9
- [Blob2007c] Blobel B: "Educational Challenge of Health Information Systems Interoperability",  
Methods Inf Med 2007, 46, S. 52-56

- [Blob2009a] Blobel B: "Analysis and Evaluation of EHR Approaches" Methods Inf Med 2009; 48, 2: pp 162-169
- [Blob2009b] Blobel B: "Approach to eHealth for enabling paradigm changes in health" Methods of Information in Medicine 2010 49 2: 123-134 Special Issue on CeHR (Methods) ISSN: 0026-1270 <http://www.schattauer.de/de/magazine/uebersicht/zeitschriften-a-z/methods/issue/special/manuscript/12617/show.html>
- [Blob2009c] Blobel B: „HL7-Standard für semantische Interoperabilität: Was kommt nach HDF und SAEAF?“ HL7-Mitteilungen, Heft 25/2009, S.9-12
- [BloGon2010] Blobel B, Gonzalez C, F. Oemig F, Lopez DM, Nykänen P, Ruotsalainen P: "The Role of Architecture and Ontology for Interoperability". In: Blobel B, Hvannberg ED, Gunnarsdottir V (Edrs.) Seamless Care – Safe Care: The Challenges of Interoperability and Patient Safety in Health Care. Proceedings of the EFMI STC 2010, pp. 33-39. Studies in Health Technology and Informatics, Vol. 155, IOS Press Amsterdam 2010, ISBN: 978-1-60750-562-5
- [BloOem2007] Blobel B, Oemig F: „Zertifizierung und Qualitätskennzeichnung zur Sicherung semantischer Interoperabilität von IT-Produkten im Gesundheitswesen“ HL7 Mitteilungen 2007; 22. Ausgabe, S.13-19, ISSN 1616-8909
- [BloOem2009] Blobel B, Oemig F: "Ontology-driven health information systems architectures" In: Adlassnig K-P, Blobel B, Mantas J, Masic I (Edrs.) Medical Informatics in a United and Healthy Europe – Proceedings of MIE 2009, pp 195-199. Series Studies in Health Technology and Informatics, Vol. 150. IOS Press, Amsterdam, Berlin, Oxford, Tokyo, Washington, IOS Press, ISBN: 978-1-60750-044-5
- [BloPha2007] Blobel B, Pharow P: "Die Rolle des EHR bei der Modernisierung des Gesundheitswesens in Deutschland" In: Blobel B, Pharow P, Zvarova J, Lopez DM (Edrs.): eHealth: Combining Health Telematics, Telemedicine, Biomedical Engineering and Bioinformatics to the Edge - CeHR Conference Proceedings 2007, pp. 343-350. IOS Press, Amsterdam, and AKA GmbH, Berlin. ISBN 978-3-89838-089-8 (Aka)
- [BodSmi2004] Bodenreider O, Smith B, Burgun A: "The Ontology-Epistemology Divide: A Case Study in Medical Terminology", Proc. FOIS-2004, Torino, Italy (2004) <http://www.lhncbc.nlm.nih.gov/lhc/docs/published/2004/pub2004064.pdf>, zuletzt zugegriffen am 14.02.10
- [CCOW] CCOW: "Clinical Context Object Working Group" <http://www.hl7.org/> -> Standards -> Clinical Context Management Specifications
- [CDA] CDA: "Clinical Document Architecture" Release (1 and) 2 <http://www.hl7.org/> -> Standards -> Clinical Document Architecture
- [CEN] CEN: Comité Européen de Normalisation: <http://www.cen.eu/cenorm/index.htm>
- [CEN TC251] CEN/TC 251: European Standardization of Health Informatics <http://centc251.org/>

- [Ceus2004] Ceusters W: Kommentar zu Harris J: "Judging the likely success of an Ontology"  
[http://www.virtualtravelog.net/entries/2004/01/judging\\_the\\_likely\\_success\\_of\\_an\\_ontology.html](http://www.virtualtravelog.net/entries/2004/01/judging_the_likely_success_of_an_ontology.html)
- [Chen2008] Chen H: "Converting RIM/CDA to Ontology for Semantic Web Reasoning"  
<http://esw.w3.org/topic/HCLS/ClinicalObservationsInteroperability/HL7CDA2OWL.html>
- [Cimi1998] Cimino J: "Desiderata for Controlled Medical Vocabularies in the Twenty-First Century"  
Methods Inf Med. 1998 Nov;37(4-5):394-403.  
[http://www.sahs.uth.tmc.edu/evbernstam/HI5300/Articles%20\(Reading%20Materials\)/cimino\\_desiderata-for-controlled-medical.pdf](http://www.sahs.uth.tmc.edu/evbernstam/HI5300/Articles%20(Reading%20Materials)/cimino_desiderata-for-controlled-medical.pdf),  
zuletzt zugegriffen am 5.5.08
- [CloMel1984] Clocksin WF, Mellish CS: „Programming in Prolog“  
Springer, 1984, ISBN: 3-450-15011-0
- [Cocc2007] Cocchiarella N: "Formal Ontology and Conceptual Realism",  
in "History and Philosophy of Logic", Dordrecht: Springer, 2007,  
ISBN 978-4020-6203-2, ISSN: 1464-5149
- [CORBA] "Common Object Request Broker Architecture"  
<http://www.corba.org/>  
s.a. [OMG]
- [CUMUL] CUMUL/LOINC: "Mehrsprachiges Verzeichnis der klinischen Analysen"  
Projekt der ELM (European Laboratory Medicine)  
<http://www.cumul.ch/>
- [D2D] D2D: Doctor to Doctor Communication  
"Die Telematik-Initiative der Kassenärztlichen Vereinigungen"  
<http://www.d2d.de/> , <http://kvno.arzt.de/mitglieder/d2d/>
- [D2D-Impl] D2D-Implementation Version 1.02, Handbuch für Softwareentwickler,  
hrsg. v. der Kassenärztlichen Vereinigung Nordrhein, Competence  
Center "IT in der Arztpraxis", Köln 2001
- [DALE-UV] DALE-UV: "Datenaustausch mit Leistungserbringern in der  
gesetzlichen Unfallversicherung"  
<http://www.dale-uv.de/>
- [DCM] "Detailed Clinical Information Models" in HL7  
<http://www.dcm.org>, <http://www.hl7.org>
- [DICOM] DICOM: "Digital Images and Communication in Medicine"  
<http://www.rsna.org/Technology/DICOM/index.cfm>
- [DICOM Supp.27] DICOM Supplement 27 – "Structured Reporting Storage SOP Classes"  
[http://61.62.116.208/webhd/adams/LinkedDocuments/sup23\\_ft.pdf](http://61.62.116.208/webhd/adams/LinkedDocuments/sup23_ft.pdf)
- [DIMDI ICD] DIMDI: ICD-10 Internationale Klassifikation der Krankheiten ,  
10.Revision  
<http://www.dimdi.de/static/de/klassi/diagnosen/icd10/index.htm>,  
zuletzt zugegriffen am 22.4.08
- [DIMDI AlphaID] DIMDI: AlphaID  
<http://www.dimdi.de/static/de/ehealth/alpha-id/index.htm>, zuletzt  
zugegriffen am 22.4.08
- [DIMDI OID] DIMDI: OID - Objekt-Identifikatoren (OIDs für das deutsche  
Gesundheitswesen)  
<http://www.dimdi.de/static/de/ehealth/oid/index.htm>

- [DIN 1463] "Erstellung und Weiterentwicklung von Thesauri; Einsprachige Thesauri"  
DIN 1463:1976-03
- [DIN 2330] "Begriffe und Benennungen; Allgemeine Grundsätze"  
DIN 2330:1979-03
- [DIN 7498] DIN ISO 7498: "Informationsverarbeitung Kommunikation Offener Systeme, Basis-Referenzmodell",  
DIN EN ISO 7498-1 ISO/OSI-Modell  
Beuth Verlag, 1982
- [Duden] Duden: "Die deutsche Rechtschreibung"  
Dudenverlag  
ISBN: 3-411-04011-4
- [Duden Inform] Duden Informatik  
Dudenverlag, 1988
- [Duden Univ] Duden – Deutsches Universalwörterbuch, 6. überarbeitete Auflage,  
Mannheim, Leipzig, Wien, Zürich: Dudenverlag 2007
- [EBCDIC] "Extended Binary Coded Decimal Interchange Code"  
<http://www.asphelper.de/referenz/EBCDIC.asp>  
<http://unicode.e-workers.de/ebcdic.php>
- [EDIFACTORY] EDIFACTORY  
<http://www.edifactory.de>, zuletzt zugegriffen am 5.5.08
- [FreiSchu2009] Freitas F, Schulz S, Moraes E: "Survey of current terminologies and ontologies in biology and medicine"  
RECIIS, Electronic Journal of Communication, Information and Innovation in Health (English edition. Online), 2009, v. 3, S.1-13
- [GanGua2001] Gangemi A, Guarino N, Masolo C, Oltramari A: "Understanding top-level ontological distinctions", Proc. of IJCAI 2001, workshop on Ontologies and Information Sharing
- [Gematik] Gematik: Gesellschaft für Telematikanwendungen der Gesundheitskarte mbH  
<http://www.gematik.de/>
- [GenMus2003] Gennari JH, Musen M, Ferguson RW, Grosso WE, Crubézy M, Eriksson H, Noy NF, Samson WT: „The evolution of Protégé: an environment for knowledge-based systems development“  
International Journal for Human-Computer Sciences 58 (2003), p.89-123
- [Gett1963] Gettier E: "Is Justified True Belief Knowledge?"  
<http://philosophy.ucsd.edu/faculty/rarneson/Courses/gettierphilreading.pdf>
- [Google] Google: "Grundlagen der Google-Suche"  
<http://www.google.com/support/websearch/bin/answer.py?hl=de&answer=35889>  
bzw. <http://www.worldsites-schweiz.ch/googles-strategie.htm>
- [Gräb2009] Gräber S: "Epidemiologie und Informatik"  
Universität des Saarlandes, 2009  
[http://www.uniklinikum-saarland.de/fileadmin/UKS/Einrichtungen/Fachrichtungen\\_Theor\\_und\\_Klin\\_Medizin/IMBEI/docs/pdf/vorlesung-epimi-WS09-Teil1.pdf](http://www.uniklinikum-saarland.de/fileadmin/UKS/Einrichtungen/Fachrichtungen_Theor_und_Klin_Medizin/IMBEI/docs/pdf/vorlesung-epimi-WS09-Teil1.pdf)
- [Grim2009] Grimm S: „Semantic Web Technologies“  
20.5.2009



- 
- [http://semantic-web-grundlagen.de/w/images/e/ee/2009-05-20\\_SWT\\_05.pdf](http://semantic-web-grundlagen.de/w/images/e/ee/2009-05-20_SWT_05.pdf), zuletzt zugegriffen am 20.8.10
- [GruWes2006] Gruber A, Westenthaler R: "Supporting domain experts in creating formal knowledge models (ontologies)"  
2006
- [Grub1992] Gruber T: "A Translation Approach to Portable Ontology Specifications"  
Knowledge Acquisition, vol.5, Issue 2, S.199-200, 1993,  
ISSN: 1042-8143
- [Grub1993] Gruber T: "Toward Principles for the Design of Ontologies used for Knowledge Sharing"  
International Journal of Human and Computer Sciences 43 (5/6),  
p.907-928, 1993
- [Grub2003] Gruber T: "It is What it Does – The Pragmatics of Ontology for Knowledge Sharing"  
CIDOC, 2003  
[http://cidoc.ics.forth.gr/docs/symposium\\_presentations/gruber\\_cidoc-ontology-2003.pdf](http://cidoc.ics.forth.gr/docs/symposium_presentations/gruber_cidoc-ontology-2003.pdf), zuletzt zugegriffen am 5.5.08
- [Grub2008] Gruber T: "Ontology"  
to appear in Encyclopedia of Database Systems, Ling Liu and M. Tamer Özsu (Eds.)  
Springer Verlag, 2008
- [Grum1974] Grumer K-W: "Beobachtung"  
B.G.Teubner, Stuttgart, 1974, ISBN: 3-519-00032-6
- [Guar1997] Guarino N: "Semantic Matching: Formal Ontological Distinction for Information Organization , Extraction, and Integration",  
in M.T. Parienza (Ed.) "Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology", 1997  
Springer Verlag, S.139-170.
- [Guar1998] Guarino N: "Formal Ontology and Information Systems"  
Proceedings of FOIS '98  
Trento, Italy, 6-8.6.98  
Amsterdam, IOS Press, pp.3-15
- [Haas2005] Haas P: "Medizinische Informationssysteme und elektronische Krankenakten",  
Springer 2005, S.223ff., ISBN 3-540-20425-3
- [Haas2006] Haas P: "Gesundheitstelematik"  
Springer Verlag, Berlin 2006
- [Hamm1997] Hammond E: "Call for a Standard Clinical Vocabulary"  
Journal American Med. Inform. Assoc. 4(3):254-255, 1997,  
<http://www.jamia.org/cgi/content/long/4/3/254>
- [Hamm2003] Hammond E: "HL7 – More than a Communications Standard"  
in "Advanced Health Telematics and Telemedicine, The Magdeburg Expert Summit Textbook", 2003  
IOS Press, S.266ff., ISBN: 1-58603-350-6
- [Harr2004] Harris J: "Judging the likely success of an Ontology", Posted to VT on January, 19<sup>th</sup>, 2004  
[http://www.virtualtravelog.net/entries/2004/01/judging\\_the\\_likely\\_success\\_of\\_an\\_ontology.html](http://www.virtualtravelog.net/entries/2004/01/judging_the_likely_success_of_an_ontology.html), zuletzt zugegriffen am 7.1.08
- [HeiDic2007] Heidenreich G, Dickmann C: "Semantic Interoperability in eHealth"  
in CeHR Conference Proceedings 2007: "eHealth: Combining Health

- Telematics Telemedicine, Biomedical Engineering and Bioinformatics to the Edge", S.89ff.  
ISBN: 978-3-89838-089-8 (Aka)
- [Hell2006] Hellmann S: "Beschreibungslogiken als Basis von OWL"  
Seminar Semantic Web Services and Interfaces, Universität Leipzig, 2006,  
<http://bis.informatik.uni-leipzig.de/de/Lehre/0607/WS/SemanticWS/files?get=sebastianhellmann.ppt>, zuletzt zugegriffen am 1.2.10
- [HIMSS] HIMSS: Interoperability Definition and Background  
[http://www.himss.org/content/files/interoperability\\_definition\\_background\\_060905.pdf](http://www.himss.org/content/files/interoperability_definition_background_060905.pdf)
- [Hinch] Hinchley A: Understanding Version 3 - A Primer on the HL7 Version 3 Communication Standard  
Alexander Moench Publishing, ISBN: 3-933819-0
- [Hirst] Hirst G: "Semantics"  
in Encyclopedia of Artificial Intelligence, Ed. Shapiro S, S.1024ff.  
Wiley Interscience, ISBN: 0-471-62974-X
- [HitKrö2008] Hitzler P, Krötzsch M, Rudolph S, Sure Y: "Semantic Web – Grundlagen"  
Springer 2008, ISBN: 978-3-540-33993-9  
<http://semantic-web-grundlagen.de/w/images/e/e7/12-owl-2-2009.pdf>, zuletzt zugegriffen am 03.02.10
- [HL7] Health Level Seven Inc. HL7. 2005,  
<http://www.hl7.org/>
- [HL7-DE] HL7-Benutzergruppe in Deutschland e.V.  
<http://www.hl7.de/>
- [HL7-DE 2007] HL7-DE: Nachrichtenprofile der HL7-Benutzergruppe in Deutschland e.V. (2007)  
<http://www.hl7.de/projektbuero/nachrichtenprofil.php>
- [HL7-Diag] HL7 Diagnoseleitfaden, v0.99e  
<http://www.hl7.de/download/documents/diagnosen/Diagnoseleitfaden-v0.99e.pdf>, zuletzt zugegriffen am 14.5.10
- [HL7-SDA] „Structured Document Architecture“  
<http://www.hl7.org/v3ballot2009may/html/welcome/introduction/index.htm>, zuletzt zugegriffen am 08.06.10
- [HL7-SMD] "Static Model Designer"  
<http://gforge.hl7.org/gf/project/static-designer/>, zuletzt zugegriffen am 08.06.10
- [HL7-SPL] "Structured Product Labeling"  
<http://www.hl7.org/implement/standards/spl.cfm>, zuletzt zugegriffen am 08.06.10  
"Implementation Guide v1.1":  
[http://www.hl7.org/search/viewSearchResult.cfm?search\\_id=98340&search\\_result\\_url=%2Fimplement%2Fstandards%2Fstandarddocument%2FSPL%20r3%20Implementation%20Guidev1%2E1%2Edoc](http://www.hl7.org/search/viewSearchResult.cfm?search_id=98340&search_result_url=%2Fimplement%2Fstandards%2Fstandarddocument%2FSPL%20r3%20Implementation%20Guidev1%2E1%2Edoc),  
zuletzt zugegriffen am 08.06.10
- [HL7 v2.x] „HL7 Version 2 Messaging Standards“  
<http://www.hl7.org/implement/standards/v2messages.cfm>, zuletzt zugegriffen am 08.06.10

- [HL7 v2.xml] Health Level Seven Inc. v2.xml Specification. 2002  
[http://www.hl7.org/Memonly/downloads/Standards\\_V2XML/XML\\_Encoding\\_Rules\\_for\\_HL7\\_v2\\_Messages.zip](http://www.hl7.org/Memonly/downloads/Standards_V2XML/XML_Encoding_Rules_for_HL7_v2_Messages.zip), zuletzt zugegriffen am 28.05.09
- [HL7-V2 Prop.DB] HL7 V2.x Proposal Database  
<http://www.hl7.org/memonly/dbsub.cfm,zuletzt> zugegriffen am 14.02.10
- [HL7-v2 Prop.605] HL7 v2.8 proposal: „Conformance Documentation Hierarchy“  
<http://www.hl7.org/documentcenter/public/wg/ictc/605%20conformance%20documentation%20v07.doc>, zuletzt zugegriffen am 08.06.10
- [HL7 V3] "HL7 Version 3",  
<http://www.hl7.org/ctl.cfm?action=ballots.home>
- [HL7-HDF] HL7-HDF: HL7 Development Framework  
HL7 Inc. HL7. 2006  
[http://www.hl7.org/library/committees/mnm%5Chdf\\_workproduct/HDF%20Methodology%20Specification%2Ezip](http://www.hl7.org/library/committees/mnm%5Chdf_workproduct/HDF%20Methodology%20Specification%2Ezip)
- [HL7-MDF] HL7-MDF: "Message Development Framework"  
Version 3.2, 1999  
<http://www.hl7.org/library/mdf99/mdf99.pdf>
- [HL7-MIF] HL7 Model Interchange Files  
<http://www.hl7.org> -> Library -> MIF
- [HL7-PubDb] HL7 Publication Database: "HL7 Tooling Collaborative Roadmap"  
[https://ohtarchitecture.projects.openhealthtools.org/wiki/HL7\\_Tooling\\_Roadmap?action=AttachFile&do=get&target=HL7+Tooling+Collaborative+Roadmap.doc](https://ohtarchitecture.projects.openhealthtools.org/wiki/HL7_Tooling_Roadmap?action=AttachFile&do=get&target=HL7+Tooling+Collaborative+Roadmap.doc), zuletzt zugegriffen am 27.05.09
- [HL7-RIM] HL7 "Reference Information Model"  
[http://www.hl7.org/library/data-model/RIM/modelpage\\_mem.htm](http://www.hl7.org/library/data-model/RIM/modelpage_mem.htm)
- [Horr2010] Horrocks I: „Description Logic: A Formal Foundation for Ontology Languages and Tools“,  
Tutorial at the "Semantic Technology Conference (SemTech)", San Francisco, CA, USA, 22.-27.6.2010  
[http://www.comlab.ox.ac.uk/ian.horrocks/Seminars/download/Horrocks\\_Ian\\_pt1.pdf](http://www.comlab.ox.ac.uk/ian.horrocks/Seminars/download/Horrocks_Ian_pt1.pdf), zuletzt zugegriffen am 8.8.10
- [H.PR.I.M] H.PR.I.M (H'): Harmoniser et PRo mouvoir les Informatique Médicales  
<http://www.hprim.org/>
- [IAO] "The Information Artefact Ontology"  
[http://www.obofoundry.org/cgi-bin/detail.cgi?id=information\\_artifact](http://www.obofoundry.org/cgi-bin/detail.cgi?id=information_artifact)  
und  
<http://purl.obolibrary.org/obo/iao.owl>
- [ICD] ICD: International Classification of Diseases  
<http://www.who.org/> und <http://www.dimdi.de/>
- [ICPM] ICPM: International Classification of Procedures in Medicine  
<http://www.who.org/>
- [ID Berlin] ID Berlin: ID-MACS  
[http://www.id-berlin.de/deu/\\_2produkte/macs.php](http://www.id-berlin.de/deu/_2produkte/macs.php)
- [IAO] Information Artefact ontology  
<http://purl.obofoundry.org/obo/iao.owl>, zuletzt zugegriffen am 10.09.09

- [IEEE] Institute of Electrical and Electronics Engineers. "IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries". New York, NY: 1990.
- [IFOMIS] Institute for Formal Ontology and Medical Information Science  
Universität des Saarlandes, Saarbrücken  
<http://www.ifomis.org/>, zuletzt zugegriffen am 23.3.09
- [IHE] IHE: Integrating the Healthcare Enterprise  
<http://www.ihe-europe.org>, <http://www.ihe-d.org>,  
<http://www.rsna.org/ihe> und <http://wiki.ihe.net>
- [IHE Vol.0] (Oemig:) IHE internal technical terminology, v0.6  
<http://www.ihe.net>, <http://groups.google.com/group/ihe-terminology/attach/7804505ee72e6ed9/IHE+Terminology+v07.zip?hl=en&part=2>, zuletzt zugegriffen am 14.2.10  
<http://www.hl7.org/documentcenter/public/wg/ictc/IHE%20Terminology%20v06.zip>, zuletzt zugegriffen am 14.2.10
- [Inge2004] Ingenerf J: Typen begrifflicher Ordnungssysteme in der Medizin  
Zeitschrift für Semiotik 2004, 26, 3-4
- [ISO 10746] ISO/IEC 10746 "Information technology – Open Distributed Processing",  
Part 2 – Reference Model.
- [ISO 17115] ISO/FDIS 17115:2007: Health informatics – Vocabulary for terminology systems
- [ISO 28379] ISO/NP TS 2837: Common Glossary for TC215  
<http://www.iso.org>
- [ISO 7498] siehe [DIN 7498]
- [ISO 8859] ISO/IEC 8859: Character Sets  
<http://www.iso.org>
- [JakÜst2007] Jakob R, Üstün B, Madden R, Sykes C: "The WHO Family of International Classifications",  
Springer Berlin / Heidelberg, 2007, S. 924-931,  
ISSN: 1436-9990 (Print), 1437-1588 (Online)
- [KBV xDT] KBV: .xDT. 2001.  
<http://www.kbv.de/> , <http://www.zi-koeln.de/> und <http://www.zi-berlin.de/>  
<http://www.kbv.de/ita/4274.html>, zuletzt zugegriffen am 3.2.10
- [Keet2009] Keet: Blog  
<http://keet.wordpress.com/2009/11/20/72010-semwebtech-lectures-34-ontology-engineering-top-down-and-bottom-up/>, zuletzt zugegriffen am 8.6.10
- [KnuFer2004] Knublauch H, Ferguson RW, Noy NF, Musen MA: "The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications"  
ISWC 2004  
<http://protege.stanford.edu/plugins/owl/publications/ISWC2004-protege-owl.pdf>, zuletzt zugegriffen am 30.8.10
- [KraMy1987] Krame BM, Mylopoulos J: "Knowledge Representation"  
in S.Shapiro "Encyclopedia of Artificial Intelligence",  
John Wiley, 1987, ISBN: 0-471-62974-X

- 
- [Künn2009] Künneth T: „Einstieg in Eclipse 3.5“  
Galileo Press  
ISBN: 978-3-8362-1428-5
- [Kusn2006] Kusnierczyk, W: „Nontological engineering“  
Formal ontology in information systems.  
In Proceedings of the 4th International Conference FOIS 2006, pp.  
39–50,  
ISBN: 0922-6389, 1-58603-685-8
- [Linke] Linke A, Nussbaumer M, Portmann PR: Studienbuch Linguistik,  
Niemeyer, Tübingen; Auflage: 5., erw. Aufl.,  
ISBN: 978-3484311213
- [LISP] LISP: LISt Processing  
Paul Graham: On Lisp, Prentice Hall 1993,  
ISBN: 0-13-030552-9
- [LOINC] LOINC: Laboratory Observation Identifiers Names and Codes  
<http://www.regenstrief.org/loinc>, <http://www.dimdi.de/klassi/loinc>,  
<http://www.loinc.de/>
- [Lopez2008] Lopez D: "Interoperable Archtiectures for Advanced Health  
Information Systems"  
Dissertation, Medizinische Fakultät der Univ. Regensburg, 22.12.09
- [Lutz2008] Lutz C: "Beschreibungslogik und Ontologiesprachen"  
Universität Bremen  
Vorlesung 2008/09  
<http://www.informatik.uni-bremen.de/tdki/lehre/ws08/bl/>, zuletzt  
zugegriffen am 1.6.10
- [Maida] Maida AS: "Frame Theory"  
in Encyclopedia of Artificial Intelligence, Ed. Shapiro S, S.300ff.  
Wiley Interscience, ISBN: 0-471-62974-X
- [Mead2005] Mead CN, Jones TM: "The Architecture of Sharing"  
Healthc Inform. 2005 Nov;22(11):35-6, 38.  
<http://www.ncbi.nlm.nih.gov/pubmed/16355527>
- [Merriam a] Merriam Webster Dictionary  
<http://www.merriam-webster.com/dictionary/interoperability>, zuletzt  
besucht am 28.3.08
- [Merriam b] Merriam Webster Dictionary  
<http://aolsvc.merriam-webster.aol.com/dictionary/vocabulary>, zuletzt  
besucht am 28.3.08
- [MiAk2004] Mika P, Akkermans H: "Towards a new synthesis of Ontology  
Technology and Knowledge Management",  
in: The Knowledge Engineering Review, Volume 19, Issue 04,  
1.Dec.2004, p.317-345.
- [Morris 1938] Morris C: "Foundations of the Theory of Signs"  
<http://www.pragmatism.org/genealogy/morris.htm>, zuletzt besucht  
am 23.3.09
- [N3] Berners-Lee T: Notation 3  
<http://www.w3.org/DesignIssues/Notation3>, zuletzt zugegriffen am  
5.3.10  
Primer: <http://www.w3.org/2000/10/swap/Primer>, zuletzt zugegriffen  
am 5.3.10  
Tutorial: <http://www.w3c.de/PubPraes/W3C->

[Standards%20und%20Semantic%20Web%20-%20Tutorial.html](#),  
zuletzt zugegriffen am 5.3.10

- [Nils1982] Nilsson NJ: "Principles of Artificial Intelligence"  
Springer, 1982, ISBN: 3-540-11340-1
- [NoyMcG] Noy N F, McGuinness D L: "Ontology Development 101: A Guide to Creating your First Ontology"  
[http://protégé.stanford.edu/publications/ontology\\_development/ontology](http://protégé.stanford.edu/publications/ontology_development/ontology)
- [Nutt1987] Nutter JT: "Epistemology",  
in S.Shapiro "Encyclopedia of Artificial Intelligence", John Wiley, 1987,  
ISBN: 0-471-62974-X
- [OBO] "The Open Biomedical Ontologies"  
<http://www.obofoundry.org/>
- [OemCre1991] Oemig F, Cremers AB, Heyer G: "Wissensbasierte Textverarbeitung: Schriftsatz und Typographie"  
DUV Informatik 1991, ISBN: 3-8244-2020-1  
<http://www.oemig.de/vieweg/default.htm>
- [OemBlo2003] Oemig F, Blobel B: "Making Messaging Standards Work: From Definition to Interoperability at Runtime"  
In: Baud R, Fieschi M, LeBeux P, Ruch P (Edrs.): The New Navigators: From Professionals to Patients, pp. 679-683. Series "Studies in Health Technology and Informatics" Vol. 95. IOS Press, Amsterdam.  
Saint Malo, France, 4-7 May 2003, <http://www.mie2003.org>  
ISBN: 1-58603-347-6
- [OemBlo2005] Oemig F, Blobel B: "Does HL7 go towards an Architecture Standard?"  
Connecting Medical Informatics and Bioinformatics.  
In: Engelbrecht R, Geisbuhler A, Lovis Ch, Mihalas G (Editors),  
Series "Studies in Health Technology and Informatics", pp.761-766,  
Volume 116, IOS Press  
Proceedings of the MIE 2005 - European Federation for Medical Informatics  
Genf, Switzerland, Aug. 2005, <http://www.mie2005.org>  
ISBN: 1-58603-549-5
- [OemBlo2006] Oemig F, Blobel B: "HL7 Version 3: Interversions Compatibility Issues"  
Ubiquity: Technologies for Better Health in Aging Societies  
In: Hasman A, Haux R, van der Lei J, De Clercq E, Roger France F (Editors)  
Series "Studies in Health Technology and Informatics, Volume 124"  
Proceedings of the MIE 2006, Proceeding CD, p.475  
Maastricht, The Netherlands  
<http://www.mie2006.org>
- [OemBlo2007a] Oemig F, Blobel B: "Semantic Interoperability Adheres to Proper Models and Code Systems: An Examination of Different Approaches or Score Systems"  
in CeHR Conference Proceedings 2007: "eHealth: Combining Health Telematics Telemedicine, Biomedical Engineering and Bioinformatics to the Edge", S.97ff.  
ISBN: 978-3-89838-089-8 (Aka)
- [OemBlo2007b] Oemig F, Blobel B: "HL7 Conformance: How to do proper messaging?"  
Medical and Care Compunetics 4  
In: Bos L, Blobel B (Editors)  
Series "Studies in Health Technology and Informatics", pp. 298-307,  
Volume 127

- Proceedings of the ICMCC 2007.  
Amsterdam, The Netherlands, IOS Press  
<http://www.icmcc2007.net>  
ISBN: 978-1-58603-71-2, ISSN: 0926-9630
- [OemBlo2009a] Oemig F, Blobel B: "Semantic Interoperability between Health Communication Standards through Formal Ontologies"  
Medical Informatics in a United and Healthy Europe.  
In: Adlassnig K-P, Blobel B, Mantas J, Masic I (Editors),  
Series "Studies in Health Technology and Informatics", pp.200-204,  
Volume 150  
Proceedings of the MIE 2009 - European Federation for Medical Informatics, IOS Press,  
Sarajevo, Bosnia and Herzegovina, 30.8-2.9.2009  
ISBN: 978-1-60750-044-5
- [OemBlo2009b] Oemig F, Blobel B: "An Ontology Architecture for HL7 V3: Pitfalls and Outcomes"  
In: Dössel O, Schlegel WC (Edrs.), IFMBE Proceedings 25/XII, p. 408 ff. Springer Verlag, Berlin, Heidelberg, New York, Tokyo.
- [OemBlo2010a] Oemig F, Blobel B. "Harmonizing the semantics of technical terms by the Generic Component Model", 10th International Special Topic Conference of the European Federation for Medical Informatics in Reykjavík Iceland, 2-4 June 2010, pp.115-121  
IOS Press, <http://www.sky.is/efmi-stc-2010-.html>,  
ISBN: 978-1-60750-562-5.
- [OemBlo2010b] Oemig F, Blobel B: "A Communication Standards Ontology using Basic Formal Ontologies"  
Medical and Care Compunetics 6  
In: Bos L, Blobel B, Benton S, Caroll D (Editors),  
Series "Studies in Health Technology and Informatics", Volume 156"  
Proceedings of the ICMCC 2010, IOS Press,  
London, U.K., 8.-11.6.2010  
<http://www.icmcc.org/>  
ISBN: 978-1-60750-564-8, p.105-113
- [OemBlo2010c] Oemig F, Blobel B: "Eine Ontologie für Kommunikationsstandards (CSO) zur Etablierung semantischer Interoperabilität zwischen HL7 v2.x und V3"  
55. GMDS-Jahrestagung  
In: Schmücker P, Ellsäßer KH, Hayna S (Hrsg.), S.584-586  
IOS Press, ISBN: 978-3-932971-11-2,  
Mannheim, 5.-9.9.2010  
<http://www.gmds2010.de/>
- [OemDud1996] Oemig F; Dudeck J.: "Problems in developing a comprehensive HL7 database"  
AMIA Fall Symposium. 1996. p.841, Hanley & Belfus Inc.  
<http://www.oemig.de/HL7/hl7amia96.htm>  
ISBN: 1-56053-208-4
- [OemPla2006] Oemig F, Platter W: „Semantische Interoperabilität - Kommunikation im Gesundheitswesen“  
Proceedings "Prozessoptimierung, eHealth und Vernetzung im Gesundheitswesen",  
Jahrbuch Gesundheitswirtschaft 2007, 5.Dez. 2006, Berlin, S.162ff  
ISBN: 3-932661-57-5
- [OemSto2000] Oemig F, Stotz L. "New Access Database Lets You Take Control of HL7 Standards"

- HL7 News Jan. 2000  
Health Level Seven, Inc.  
<http://www.hl7.org/>, <http://www.oemig.de/HL7/>, zuletzt zugegriffen am 1.9.09
- [OemThu2007a] Oemig F, Thun S: "Implementation Guide: Score-Systems and Assessments"  
TCs der HL7 Benutzergruppe in Deutschland, 2007
- [OemThu2007b] Oemig F, Thun S: "Übertragung von Scores und Assessments mittels HL7 Version 3"  
GMDS Plenary Meeting: "Medizin und Gesellschaft; Prävention und Versorgung: innovativ, qualitätsgesichert, sozial"  
17.-21.Sept. 2007, Augsburg, S.257  
ISBN: 3-938975-12-1
- [OemVei2000] Oemig F; Veil K: "HL7 V.2.4 Styleguide"  
HL7 Winter Working Group Meeting 2000  
24.-29.Jan. 2000, San Diego, CA, USA
- [Ogde1923] Ogden CK, Richards IA: "The Meaning of Meaning: A Study of the Influence of Language Upon Thought and of the Science of Symbolism",  
Routledge & Kegan Paul, London 1923.
- [OMG] Object Management Group, Inc.: CORBA Specifications.  
<http://www.omg.org/>
- [Oost2000] Oosterwijk H.: "DICOM Basics"  
OTech Inc., Cap Gemini Ernst & Young. 2000. ISBN: 90-75498-39-X  
Second Edition: ISBN: 0-9718867-0-9
- [OPPL] "Ontology Preprocessor Language"  
<http://oppl.sourceforge.net/>, zuletzt zugegriffen am 14.02.10  
<http://www.cs.manchester.ac.uk/~iannonel/oppl/>, zuletzt zugegriffen am 14.02.10
- [Orgu2002] Orgun B: "HL7-RIM Ontology as a Protégé project"  
<http://www.ics.mq.edu.au/~borgun/RIM.zip>  
<http://web.science.mq.edu.au/~borgun/RIM-HL7All.zip>  
Diskussion: <http://osdir.com/ml/w3c.working-groups.swhcls/2006-02/msg00071.html>
- [OWL] OWL: "Web Ontology Language"  
W3C Recommendation 10. February 2004  
<http://www.w3.org/TR/owl-features/>, zuletzt zugegriffen am 5.5.08
- [OWL2] OWL2: Web Ontology Language  
Primer: <http://www.w3.org/TR/owl2-primer/>, zuletzt zugegriffen am 10.06.10
- [OWL2 Profile] OWL2 Language Profiles  
W3C: <http://www.w3.org/TR/owl2-profiles/>, zuletzt zugegriffen am 10.6.10
- [OWLPropViz] „OWL Viz“, Plug-In für Protégé,  
[http://protegewiki.stanford.edu/wiki/OWLViz\\_build\\_22](http://protegewiki.stanford.edu/wiki/OWLViz_build_22), zuletzt besucht am 30.8.10  
<http://www.co-ode.org/downloads/owlviz/>, zuletzt besucht am 30.8.10  
GraphViz wird für OWLViz benötigt: für Protege 4.0.2 wird die OWLViz Version 4.1 benötigt:  
<http://protege.stanford.edu/plugins/protege4/org.coode.owlviz.jar>,  
zuletzt besucht am 30.8.10



- 
- [PaDok] Fraunhofer-Institut für Biomedizinische Technik: PaDok  
<http://www.ibmt.fhg.de/>
- [Pellet] Pellet, Open Source Reasoner  
<http://pellet.owldl.com/>, zuletzt zugegriffen am 5.5.08
- [PROMPT] Tool for ontology mapping  
<http://protege.stanford.edu/plugins/prompt/prompt.html>
- [Protégé] Protégé: The Protégé Project  
<http://protege.stanford.edu/>, zuletzt zugegriffen am 5.5.08
- [Quine1948] Quine W.V.O.: „On what there is“, From a Logical Point of View, Cambridge, Mass., 1953, 1960, 1980: Harvard University Press, 1-19, 1948
- [RACER] Racer Systems GmbH & Co KG  
<http://www.racer-systems.com/products/tools/protege.phtml>, zuletzt zugegriffen am 5.5.08
- [RDF] RDF: "Resource Description Framework"  
<http://www.w3.org/RDF/>, zuletzt zugegriffen am 5.5.08
- [RDFS] RDFS: "Resource Description Framework Schema"  
W3C Candidate Recommendation 27.3.00  
<http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>, zuletzt zugegriffen am 17.3.09
- [Rect2008] Rector A: "Barriers, approaches and research priorities for integrating biomedical ontologies", Information Society and Media, 2008.  
[www.semantichhealth.org/DELIVERABLES/SemanticHEALTH\\_D6\\_1.pdf](http://www.semantichhealth.org/DELIVERABLES/SemanticHEALTH_D6_1.pdf), zuletzt zugegriffen am 29.05.10
- [RM-ODP] International Organization for Standardization (1996) ISO/IEC 10746-2. Information technology – "Reference Model for Open Distributed Processing": Foundations. Geneva. ISO/IEC IS 10746 / ITU-T x.901, ISO/IEC IS 10746 / ITU-T x.901  
<http://www.rm-odp.net/>, zuletzt zugegriffen am 17.3.09
- [SAEAF] HL7: "Services-Aware Enterprise Architecture Framework for HL7 (v0.8)", Vancouver Draft, 20.09.08  
<http://wiki.hl7.org/images/3/3a/SAEAF.zip> (User: wiki, Pwd: wikiwiki), zuletzt zugegriffen 24.2.09  
<http://gforge.hl7.org/gf/project/saeaf/>, zuletzt zugegriffen am 15.05.10
- [SAP] SAP: HCM-Spezifikation  
<http://ifr.sap.com/>
- [SAP HCM] SAP: "IS-HCM-Handbuch für externe Systempartner"  
August 1999 (IS-H-Release 4.03)
- [SchKus2007] Schober D, Kusnierczyk W, Lewis SE, Lomax J, Mungall C, Rocca-Serra P, Smith B, Sansone S-A: "Towards naming conventions for use in controlled vocabulary and ontology engineering"  
[http://nemo.nic.uoregon.edu/wiki/images/1/1f/OBO\\_NamingConventions2007.pdf](http://nemo.nic.uoregon.edu/wiki/images/1/1f/OBO_NamingConventions2007.pdf), zuletzt zugegriffen am 20.05.10
- [SchoOem2001] Scholz P, Oemig F. "HL7 - Content Management for Standards"  
HL7 2nd Int`l Affiliate Meeting  
31.Aug.-1.Sep. 2001, Reading, UK
- [SchuBei2009] Schulz S, Beisswanger E, van den Hoek L, Bodenreider O, van Mulligen EM: "Alignment of the UMLS semantic network with BioTop:

- methodology and assessment",  
Bioinformatics 2009 25(12):i69-i76;  
doi:10.1093/bioinformatics/btp194
- [SchuKum2006] Schulz S, Kumar A, Bitter T: "Biomedical ontologies: what part-of is and isn't"  
Special issue: Biomedical ontologies  
Journal of Biomedical Informatics, Volume 39 , Issue 3 (June 2006),S.350-361  
ISSN: 1532-0464
- [SchulzvT] Schulz von Thun, F: "Kommunikationsquadrat"  
<http://www.schulz-von-thun.de/mod-komquad.html>, zuletzt  
zugegriffen am 22.4.08
- [SCIPHGX] SCIPHGX: Standardisation of Communication between Information  
Systems in Physician Offices and Hospital Using XML.  
Working Draft 15  
<http://sciphox.hl7.de/>
- [SGML] SGML: "Standard Generalized Markup Language"  
<http://www.w3.org/MarkUp/SGML/>
- [Shor1976] Shortliffe EH: "Computer-based Medical Consultations, MYCIN"  
Elsevier, 1976, ISBN 0-444-00179-4
- [SinJuu2001] Singureanu I, Juurlink M A: Conformance Tutorial  
Jan. 2001  
<http://www.hl7.org/Library/Committees/Conf/ConformanceTutorial%20Eppt>, zuletzt zugegriffen am 5.5.08
- [Sing2003] Singureanu I: "Making Version 3 Implementations 'Version 3 Ready'  
Paves the Way to Successful Migration"  
20.10.2003  
<http://www.hl7.org/Library/Committees/Conf/V2%20DV3%20Migration%20Epdf>, zuletzt zugegriffen am 5.5.08
- [SmiCeu2005] Smith B, Ceusters W, Klagges B, Köhler J, Kumar A, Lomax J, Mungall C, Neuhaus F, Rector AL, Rosse C.: "Relations in biomedical ontologies"  
Genome Biology 2005;6(5):R46. Epub 2005 Apr 28.
- [SmiCeu2006] Smith B, Ceusters W: "HL7 RIM: An Incoherent Standard"  
Ubiquity: Technologies for Better Health in Aging Societies  
In: Hasman A, Haux r, van der Lei J, De Clercq E, Roger France F (Editors)  
Series "Studies in Health Technology and Informatics, Volume 124"  
Proceedings of the MIE 2006, Proceeding CD, p.133-138  
Maastricht, The Netherlands  
<http://www.mie2006.org>
- [Smit2001] Smith B.: "Ontology and Information Systems"  
[http://ontology.buffalo.edu/ontology\(PIC\).pdf](http://ontology.buffalo.edu/ontology(PIC).pdf)  
zuletzt zugegriffen am 29.5.10
- [Smit2004] Smith B: "Beyond Concepts: Ontology as Reality Representation"  
Proceedings of the International Conference on Formal Ontology in Information Systems, 11, 2004, S.39-50
- [SmiWel2001] Smith B, Welty C: "Ontology: Towards a New Synthesis"  
FOIS'01, October 17-19, 2001, Ogunquit, Maine, USA. S.3-9  
<http://lists.w3.org/Archives/Public/www-webont-wg/2002Aug/att-0056/fois-intro.pdf>, zuletzt zugegriffen am 4.2.10

- 
- [Snomed] SNOMED. Systematized NOMenclature of MEDicine  
<http://www.snomed.org/>, <http://www.ihtsdo.org>
- [Snomed CT] SNOMED CT: SNOMED Clinical Terms  
<http://www.snomed.org/>, <http://www.ihtsdo.org>
- [SolChu2009] Solbrig H, Chute C: „Concepts, Modeling and Confusion“  
International Conference of Biomedical Ontologies, Buffalo, NY, ICBO  
2009  
<http://icbo.buffalo.edu/Proceedings.pdf#page=137>, zuletzt  
zugegriffen am 15.06.10
- [SPARQL] SPARQL Query Language for RDF  
<http://www.w3.org/TR/rdf-sparql-query/>, zuletzt zugegriffen am  
5.4.10
- [SteGob2000] Stevens R, Goble CA, Bechhofer S: "Ontology-based Knowledge  
Representation for Bioinformatics"  
Briefings in Bioinformatics, 1 (4) 2000, S.398-414
- [SWRL] SWRL: "A Semantic Web Rule Language combining OWL and RuleML"  
Mai 2004  
<http://www.w3.org/Submission/SWRL/>, zuletzt zugegriffen am  
10.6.10  
<http://protege.stanford.edu/conference/2009/slides/SWRL2009ProtegeConference.pdf>, zuletzt zugegriffen am 1.6.10
- [TeVeGe] TeVeGe: Gesellschaft für ein vernetztes Gesundheitswesen GmbH  
<http://www.tevege.de/>
- [Thom2010] Thommen J-P: „Pragmatismus“  
Gabler Verlag (Herausgeber), Gabler Wirtschaftslexikon, Stichwort:  
Pragmatismus,  
<http://wirtschaftslexikon.gabler.de/Archiv/12511/pragmatismus-v5.html>, zuletzt zugegriffen am 2.3.10
- [ThuOem2007] Thun S, Oemig F, Rüth R, Röhrig R: "Abbildung von  
intensivmedizinischen und anästhesiologischen Scores und  
Assessments mittels LOINC und SNOMED CT"  
GMDS Plenary Meeting  
Sept. 2007, Augsburg
- [TI] Telematik-Infrastruktur  
[http://www.gematik.de/\(S\(1dvvmhmacjhobf3kiaqn1va0\)\)/Telematik.Gematik](http://www.gematik.de/(S(1dvvmhmacjhobf3kiaqn1va0))/Telematik.Gematik), zuletzt zugegriffen am 1.8.08
- [TOG2009] The Open Group: SOA Ontology, Draft 3.2, 2009
- [Tu2008] Tu S: HL7 RIM Ontology  
<http://protege.stanford.edu/ontologies/HL7RIM/index.html>  
(<http://protege.stanford.edu/ontologies/HL7RIM/HL7RIM.zip>)  
zuletzt zugegriffen am 09.08.09
- [UN/EDIFACT a] UN/EDIFACT: United Nations Directories for Electronic Data  
Interchange for Administration, Commerce and Transport  
<http://www.unece.org/trade/untdid/welcome.htm>
- [UN/EDIFACT b] UN/EDIFACT: D.04A (Batch), Syntax version 3  
<http://www.gefeg.com/edifact/Index.htm>
- [Unicode] Unicode Consortium  
<http://www.unicode.org/>, zuletzt zugegriffen am 17.3.09  
<http://unicode.e-workers.de/>, zuletzt zugegriffen am 17.3.09

- [Uscho1996] Uschold M: "Building Ontologies: Towards a unified methodology", AIAI Technical Report September 1996 (AIAI-TR-197), Edinburgh 1996.
- [UschGru1996] Uschold M, Gruninger M: "Ontologies: Principles, Methods and Applications"  
The Knowledge Engineering Review, 11(2), p.93-136, 1996
- [UschGru2004] Uschold M, Gruninger M: "Ontologies and Semantics for Seamless Connectivity",  
ACM SIGMOD Record 33(4), 2004: p.58-64,  
ISSN:0163-5808
- [VCS] VCS: VdAP Communication Standard  
Beuth Verlag GmbH, Berlin, PAS 1011:2001-03, Juli 2002  
<http://www.vdap.de/>
- [Wach2008] Wachsmann L: „Entwurf und Implementierung eines Modells zur Visualisierung von OWL-Properties als Protégé-PlugIn mit Layoutalgorithmen aus Graphviz“, Diplomarbeit an der TU Braunschweig, 2008  
<http://www.wachsmann.tk/owlpropviz/>, zuletzt besucht am 30.8.10  
<http://www.wachsmann.tk/owlpropviz/diplomarbeit-wachsmann.pdf>,  
zuletzt besucht am 30.8.10
- [WalRis1999] Walker, Rishel, Beeler, Schadow, Huff, et.al.: "Message Development Framework", Version 3.2 - MDF99  
HL7 Inc. HL7, 1999  
<http://www.hl7.org/Library/Committees/mnm/Mdf99%2Epdf>
- [Wern2001] Wernke et.al. (Hrsg.): Duden Fremdwörterbuch  
Mannheim, Leipzig, Wien, Zürich  
Dudenverlag 2001  
ISBN: 978-3-411-04059-9
- [WesScha2005] Westenthaler R, Schaffert S, Gruber A: "A Semantic Wiki for Collaborative Knowledge Formation"  
in S. Reich et.al. (Eds.): "Semantic Content Engineering", Proceedings der "Semantics 2005", Trauner Verlag.
- [WSDL] WSDL: "Web Services Description Language (WSDL) 1.1"  
März 2001  
<http://www.w3.org/TR/wsdl>, zuletzt zugegriffen am 5.5.08
- [WHO] World Health Organization: "Manual of the International Classification of Diseases, Injuries,, and causes of Death",  
Geneva, Switzerland, 1977
- [XML] XML: Extensible Markup Language 1.0, 2nd Edition W3C Recommendation  
<http://www.w3.org/TR/REC-xml>
- [XML Schema] XML Schema, W3C Recommendation May 2, 2001  
<http://www.w3.org/XML/Schema>, <http://www.w3.org/TR/xmlschema-0/>,  
<http://www.w3.org/TR/xmlschema-1/>,  
<http://www.w3.org/TR/xmlschema-2/>
- [Yahoo] Yahoo: Bereiche auf der Suchergebnisseite  
<http://help.yahoo.com/l/de/yahoo/search/allgemein/basics-23.html>,  
zuletzt besucht am 17.03.09
- [Zaiß] Zaiß A, et.al.: "Medizinische Dokumentation und Terminologie und Linguistik"

in Lehmann "Handbuch der medizinischen Informatik", S.90ff.,  
Hanser Verlag, ISBN: 3-446-22701

## 22.5. Index

Index	Seite	Index	Seite
<b>-§-</b>			
§301 .....	15	CMET .....	90
<b>-A-</b>			
A-Box .....	44, 46, 106, 152	CNE .....	65
ACGT .....	74, 119, 124, 154	Computational View .....	33
Versionierung .....	154	CORBA .....	167
Agent .....	43	CSO	
Akteur .....	15, 19	Ergänzung .....	134
Alphabet .....	51	CWE .....	65
Alpha-ID .....	30	<b>-D-</b>	
ambulanter Bereich .....	7	D2D .....	15
Apotheken .....	9	DALE-UV .....	15
ARB .....	63	DatatypeProperty .....	49, 87, 108, 155
Architektur .....	21	Dateien	
ASTM .....	15	Verzeichnisstruktur .....	158
Aufgabenstellung .....	11	Delimiter .....	62, 67
<b>-B-</b>			
Basiskonzepte .....	83, 130	DICOM .....	85
Basisprinzipien des GCM .....	38	D-MIM .....	39, 58, 90
Begriffsmenge .....	50	Dreieck	
Beobachtungseinheit .....	50	semiotisches .....	24
Beschreibungslogik .....	132	<b>-E-</b>	
Best of Breed .....	8	eBXML .....	15
Bewertung der Arbeit		EDIFACT .....	15
Vor-/Nachteile .....	166	Engineering View .....	33
vorher/nachher .....	165	Enterprise View .....	33
Beziehung .....	29, 50	Entity-Relationship-Modell .....	28
assoziativ		Epistemologie .....	26
Spezialfälle .....	54	ER7 .....	62
BFO .....	76, 77, 118, 124	Ereignis .....	14
<b>-C-</b>			
CDA .....	39	Ergebnisse	
Rel.1 .....	39	Datei-Download .....	160
		Erkenntnistheorie .....	26
		Expression	
		regular .....	136, 140
<b>-F-</b>			
		Fachdomäne .....	74, 109

First-Order-Logik .....	47	<b>-J-</b>	
Frames .....	28, 53	Java .....	149
<b>-G-</b>		<b>-K-</b>	
GCM .....	29, 32, 42, 172	Klassifikation .....	30
HL7 v2.x .....	97	Kommunikation .....	8
HL7 V3 .....	88	intersektoral .....	9
GCS .....	129	Kommunikationsserver	
Glauben .....	27	Konfiguration .....	106
Gliederung der Arbeit .....	12	Komponentenmodell	
Google .....	174	generisch .....	32
<b>-H-</b>		Konsistenzprüfung .....	155
HDF .....	58	Konsistenzregeln .....	108
HermiT .....	48	Konzept .....	28
Hierarchiebildung		<b>-L-</b>	
Beziehung .....	44	Länge	
HL7 .....	11	minimal/maximal .....	67
HL7 v2.x		Leistungserbringer .....	7
PID-Segment .....	105	LISP .....	48
HL7 V3 Tooling .....	88	LOINC .....	51
HMD .....	90	<b>-M-</b>	
<b>-I-</b>		Mapping .....	133, 135, 158
ICD-Codes .....	30	Regelwerk .....	130
ICT .....	8	MDF .....	58
ID MACS .....	30	Merkmale .....	50
IFOMIS		MIF .....	88, 91, 149, 180, 181
ACGT .....	119	cmetList.mif .....	199
BFO .....	118	DEFN=UV=DT=1.0.coremif .....	197
Import .....	153	Fehler .....	96
Include .....	153	Namensgebung .....	96
Information Artefact Ontology .....	75	RIM.coremif .....	181
Information Object Quality .....	83	R-MIM.mif .....	189
Information View .....	33	vocabulary.mif .....	185
Informationszyklus .....	11, 20	Modell .....	21
Instanz .....	87	Motivation .....	9
Interoperabilität .....	21	MYCIN .....	48
informelle Definition .....	8	<b>-N-</b>	
ISDN .....	15	N3 .....	80
ISO/OSI-Modell .....	11, 18	Nachrichtenaustausch .....	8
ITS .....	90		

Namenskonvention .....	158	Pragmatismus .....	25
Namespace .....	85, 154	Prolog .....	48
Nominalphrasen .....	24	Protégé .....	48, 91
		3.4 .....	156
<b>-O-</b>		Fehler.....	156
ObjectProperty	49, 72, 78, 84, 87, 104, 106, 108, 128	4.0.2 .....	156
OCL .....	170	Fehler.....	156
OMG .....	167	4.1β .....	156
Ontologie .....	25	4.1α .....	156
Fachdomäne .....	124	Basis-Ontologie.....	77
Referenz .....	74	Core .....	172
Struktur.....	73	Prozesse	
Top-Level.....	44	Kommunikation und Information .....	7
Typen .....	44	Pub-DB .....	88, 149
OPPL.....	90, 180	Punkte	
Ordnungsstruktur		offene.....	170
semantisch.....	50		
syntaktisch.....	51	<b>-Q-</b>	
OSGi.....	172	Qualitäten .....	83
OWL .....	49, 70, 176		
Beispiel.....	138	<b>-R-</b>	
Generator		RACER .....	48
HL7 v2.x.....	152	RDF .....	48, 49, 70, 174
HL7 V3 .....	153	Beispiel .....	175
Generierungsaspekte.....	149	Reasoner.....	43, 49
HL7 v2.x .....	97	Fact++ .....	155
HL7 V3 .....	87, 90	Pellet.....	155
Import.....	153	Reasoning .....	48, 91, 108, 129, 155
Scores .....	127	Referenzmodell .....	32
ACGT .....	129	Referenz-Ontologie .....	80
Version 1 .....	178	Regeln .....	53
OWL 2.....	179	RegularExpression .....	136
		regular Expression.....	132, 140
<b>-P-</b>		Relation.....	29, 86
Partonomie .....	53	is-a .....	51
Pellet .....	48	part-of.....	51
PEO .....	16	Relationen	
Persistenzstrukturen .....	167	erste Übersicht .....	72
Petri-Netze .....	28	Relationship Ontology .....	78
Prä-/Post-Koordination .....	30	RIM .....	58
Pragmatik.....	25	Fehler.....	96
		R-MIM.....	90



Fehler .....	96	Datenbank .....	168
RM-ODP .....	32	HL7 v2.x .....	59
RO .....	86	HL7 V3 .....	68
		Scores .....	126
<b>-S-</b>		<b>-V-</b>	
SAEAF .....	34, 58	VBA .....	149
Schlussfolgerung .....	172	VB-Script.....	149
Scripting .....	149	VCS .....	15
SDO .....	16	Verarbeitung	
Sektor.....	9	natürlichsprachlich .....	24
Semantik.....	23	Versicherungen .....	7
semiotisches Dreieck .....	24	Versorgung	
Snomed CT .....	51, 71	ambulant .....	9
stationärer Bereich .....	7	integriert.....	9
Stereotypen.....	53	stationär .....	8
SWRL.....	109, 180	Vocabulary Domain.....	153
Synonyme .....	54	Vokabular.....	51
Syntax .....	23		
<b>-T-</b>		<b>-W-</b>	
Taxonomie.....	53	Webservice .....	168
T-Box.....	87, 106	WHO .....	30
Technology View .....	33	Wissen .....	10, 21, 27
Terminologie.....	52	Wissensbasis.....	28
Thesaurus .....	52	Wissenspyramide.....	27
TI 9		Wissensrepräsentation .....	28
Top-Level-Ontologie.....	44	WSDL.....	168
Trennsymbole .....	62		
Truncation .....	67	<b>-X-</b>	
		XSLT .....	149
<b>-U-</b>		<b>-Y-</b>	
UML		Yahoo .....	174
Begrifflichkeiten .....	53	<b>-Z-</b>	
Diagramm .....	28	Zusammenfassung .....	172
Klassendiagramm.....	53, 58		
UML-Modell			
Business-Objekte .....	167		