

---

**TERMINOLOGISCHE FUNDIERUNG VON  
DOKUMENTATIONSSYSTEMEN  
– EIN BEITRAG ZUR VERNETZTEN FORSCHUNG  
UND INTEGRIERTEN VERSORGUNG –**

**Sylvie Marie-Noel Ngouongo Nguemegne**

---



München 2015



Medizinische Fakultät  
der  
Ludwig-Maximilian-Universität München

Aus dem Institut für Medizinische Informationsverarbeitung, Biometrie und Epidemiologie  
(IBE)

Vorstand: Univ. Prof. Dr. Ulrich Mansmann

TERMINOLOGISCHE FUNDIERUNG VON DOKUMENTATIONSSYSTEMEN  
– EIN BEITRAG ZUR VERNETZTEN FORSCHUNG  
UND INTEGRIERTEN VERSORGUNG –

Dissertation  
zum Erwerb des Doktorgrades  
der Humanbiologie (Dr. hum. biol.)  
an der Medizinischen Fakultät der  
Ludwig-Maximilians-Universität zu München

vorgelegt von  
Sylvie Marie-Noel Nguoungo Nguemegne  
aus Kamerun  
2015

**Mit Genehmigung der Medizinischen Fakultät  
der Universität München**

Berichterstatter: Prof. Dr. Jürgen Stausberg

Mitberichterstatter: Prof. Dr. Eva Grill  
Prof. Dr. Dr. Stefan Wirth

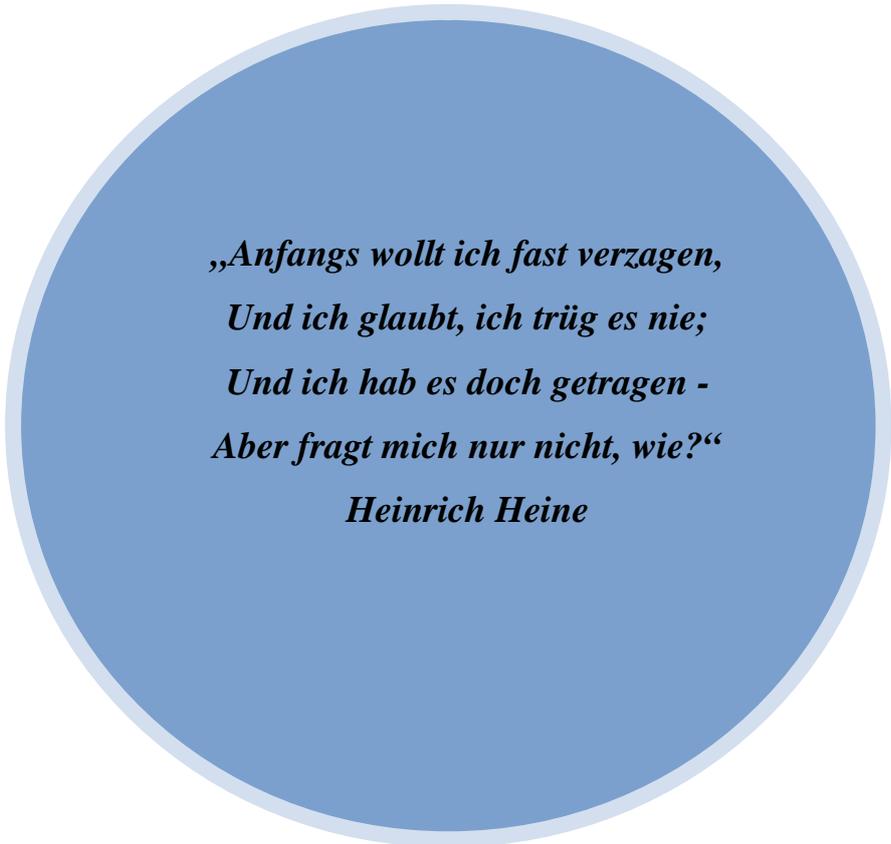
Mitbetreuung durch den  
promovierten Mitarbeiter:

Dekan: Prof. Dr. med. dent. Reinhard Hickel

Tag der mündlichen Prüfung: 11.12.2015

### **Publikationen im Umfeld dieser Arbeit**

- [1] S. MN Ngouongo, J. Stausberg, Integration of Classifications and Terminologies in Metadata Registries Based on ISO/IEC 11179, Stud Health Technol Inform 169 (2011) 744-748.
- [2] S. MN Ngouongo, J. Stausberg, Harmonisierung von Datenbeständen der Medizin - Ein Weg zu besserer Qualität und höherer Effizienz empirischer Forschung, MDI 2011, 13(3): 129-132.
- [3] S. MN Ngouongo, M. Löbe, J. Stausberg, The ISO/IEC 11179 norm for metadata registries: does it cover healthcare standards in empirical research? J Biomed Inform 46(2)(2013) 318-327.
- [4] S. MN Ngouongo, J. Stausberg, Terminology-based documentation systems: a systemantic comparison of four different approaches, Stud Health Technol Inform 190 (2013) 97-99.



*„Anfangs wollt ich fast verzagen,  
Und ich glaubt, ich trüg es nie;  
Und ich hab es doch getragen -  
Aber fragt mich nur nicht, wie?“*

*Heinrich Heine*

# INHALTSVERZEICHNIS

<b>A</b>	<b>Einleitung</b> .....	<b>1</b>
A.1	GEGENSTAND UND MOTIVATION .....	2
A.2	PROBLEMSTELLUNG .....	4
A.3	ZIELSETZUNG.....	5
A.4	AUFGABENSTELLUNG.....	6
A.5	AUFBAU DER ARBEIT .....	7
<b>B</b>	<b>Grundlagen</b> .....	<b>9</b>
B.1	GRUNDLAGEN DER MEDIZINISCHEN DOKUMENTATION .....	9
B.1.1	<i>Allgemeines</i> .....	9
B.1.2	<i>Medizinische Dokumentation und medizinische Dokumentationssysteme</i> .....	9
B.1.3	<i>Objekte, Merkmale, Merkmalskataloge</i> .....	12
B.1.4	<i>Medizinische kontrollierte Vokabulare</i> .....	13
B.1.5	<i>Abschnittszusammenfassung</i> .....	16
B.2	TECHNISCHE GRUNDLAGEN .....	18
<b>C</b>	<b>Material und Methoden</b> .....	<b>23</b>
C.1	VORGEHEN .....	23
C.2	TOP-DOWN-ANSATZ.....	24
C.3	BOTTOM-UP-ANSATZ.....	26
C.3.1	<i>TERMTrial (Terminology-based Remote-Data-Entry-System for clinical trials)</i> .....	26
C.3.2	<i>InCoMe (Integriertes System zur Computerunterstützten Entwicklung medizinischer Dokumentationssysteme)</i> .....	30
C.3.3	<i>ORCA (Open Record for CARE)</i> .....	35
C.3.4	<i>ICNP-based ENRS (ICNP-based Electronic Nursing Record System)</i> .....	37
C.4	ANALYSE- UND QUALITÄTSKRITERIEN .....	39
<b>D</b>	<b>Analyse bestehender Ansätze</b> .....	<b>41</b>
D.1	EINFÜHRUNG .....	41
D.2	TOP-DOWN-ANSATZ: ISO/IEC 11179 ED.3 .....	41
D.2.1	<i>Nutzung eines kontrollierten Vokabulars (terminologische Fundierung)</i> .....	41
D.2.2	<i>Merkmalsebene</i> .....	42
D.2.3	<i>Wertebereiche</i> .....	42
D.2.4	<i>Rollen und Entitätstypen</i> .....	42
D.2.5	<i>Einschränkungen</i> .....	43
D.3	BOTTOM-UP-ANSATZ.....	46
D.3.1	<i>TERMTrial</i> .....	46
D.3.2	<i>InCoMe</i> .....	47

D.3.3	ORCA.....	48
D.3.4	ICNP-based ENRS.....	50
D.4	ZUSAMMENFASSUNG DER ERGEBNISSE.....	52
<b>E</b>	<b>Modellentwicklung.....</b>	<b>55</b>
E.1	GRUNDIDEE .....	55
E.2	SPEZIFIKATION VON VOKADOKS .....	56
E.2.1	Einführung.....	56
E.2.2	Konzept und Design.....	57
E.2.3	Das VokaDoks klinische Informationsmodell.....	58
E.2.4	Nutzung eines kontrollierten Vokabulars (terminologische Fundierung).....	59
E.2.5	Merkmalsebene.....	64
E.2.6	Wertebereiche.....	65
E.2.7	Rollen und Entitätstypen.....	66
E.2.8	Constraints .....	66
E.2.9	Abschnittszusammenfassung.....	67
<b>F</b>	<b>Prototypische Implementierung.....</b>	<b>69</b>
F.1	ARCHITEKTUR UND PROGRAMMIERUNG.....	69
F.1.1	Allgemeines .....	69
F.1.2	Grundlegende Aspekte.....	69
F.1.3	Abbildung von InCoMe in ISO/IEC 11179 Ed.3 .....	70
F.1.4	InCoMe-Ausbau.....	72
F.1.5	Technische Anforderungen und Entwicklungsumgebung.....	73
F.1.6	Einschränkungen.....	76
F.2	SYSTEMBESCHREIBUNG.....	78
F.2.1	Einführung.....	78
F.2.2	Nutzung von kontrollierten Vokabularen .....	78
F.2.3	Erfassung von Merkmalsarten/Data Elements (Merkmalsebene) .....	79
F.2.4	Erfassung von Stammdaten (Wertebereiche) .....	80
F.2.5	Bündelung von Kontexten/Data_Element_Concept .....	84
F.2.6	Definition von Integritätsbedingungen (Constraints).....	86
F.2.7	Sicherung der Modellierung.....	91
F.2.8	Export des Dokumentationsmodells .....	91
F.2.9	Einschränkungen.....	96
F.3	DAS VOKADOKS-BASIERTE DOKUMENTATIONSSYSTEM.....	97
F.3.1	Einführung.....	97
F.3.2	Installation eines Anwendungsbeispiels des VokaDoks-medizinischen Dokumentationssystems.....	97

<b>G</b>	<b>Diskussion .....</b>	<b>107</b>
<b>H</b>	<b>Ausblick.....</b>	<b>111</b>
<b>I</b>	<b>Zusammenfassung.....</b>	<b>113</b>
<b>J</b>	<b>Literaturverzeichnis.....</b>	<b>115</b>
<b>K</b>	<b>Anhang .....</b>	<b>121</b>
	K.1 MODELLDIAGRAMME .....	121
	K.2 VOKADOKS-MODELLBESCHREIBUNG .....	128
	<i>K.2.1 Modellierungsablauf mit dem VokaDoks-Designer .....</i>	<i>128</i>
	<i>K.2.2 SQL-Datenbankskript für die Erzeugung der Merkmalskatalog-Datenbank.....</i>	<i>129</i>
	<i>K.2.3 Anwendung appl.xml .....</i>	<i>133</i>
	<i>K.2.4 Anwendungskonfiguration web.xml.....</i>	<i>142</i>
	<i>K.2.5 CDMS-Source code und Anwendung .....</i>	<i>145</i>
	<b>Danksagung.....</b>	<b>147</b>



## VERZEICHNIS DER ABBILDUNGEN

Abbildung 1: Grobe Darstellung des <i>Data Description Package</i> mit Beispielen in grau [31] .....	26
Abbildung 2: Systemarchitektur von TERMTrial [47] .....	29
Abbildung 3: Form Building Component [47].....	30
Abbildung 4: Dokumentationsmodell InCoMe [20] .....	34
Abbildung 5: Architektur InCoMe [20] .....	35
Abbildung 6: Beispiel des Inhalts der Knowledge Base [18].....	37
Abbildung 7: Pflegeinformationsmodell für das elektronische Pflegedokumentationssystem (ENRS) [55] .....	39
Abbildung 8: Einbettungsmöglichkeit von ICD-10-GM in die <i>Concept System Metamodel Region</i> des ISO/IEC 11179 Ed.3 <i>Concepts Package</i> .....	44
Abbildung 9: Erzeugung eines Dokumentationsmerkmals im <i>Data Description Package</i> des ISO/IEC 11179 Ed.3 .....	44
Abbildung 10: Darstellungsmöglichkeit eines <i>Data_Element</i> auf einem elektronischen Case Report Form (eCRF). .....	45
Abbildung 11: Modellelemente mit Beispielen zur Dokumentation von Informationen im Dokumentationssystem (CDMS) von InCoMe .....	48
Abbildung 12: Beispiel der Erfassung des „Level“ des „Observe Consciousness“ eines Patienten im ENRS.....	51
Abbildung 13: UML-Hauptanwendungsfälle als Arbeitsschritte/-pakete zur Entwicklung eines terminologisch fundierten Dokumentationssystems.....	56
Abbildung 14: Architektur des medizinischen Vokabular-basierten Dokumentationssystems (VokaDoks) mit Interaktionen zwischen seinen verschiedenen Modulen .....	57
Abbildung 15: Klinisches Informationsmodell für das Vokabular-basierte klinische Dokumentationssystem VokaDoks .....	59
Abbildung 16: UML Klassendiagramm des Moduls <i>Controlled Vocabularies</i> .....	61
Abbildung 17: UML Klassendiagramm des Moduls <i>Statement</i> zur Erarbeitung und Verwaltung von präkoordinierten medizinischen Begriffen in VokaDoks.....	62
Abbildung 18: UML Klassendiagramm zur Darstellung der Nutzung von <i>Links</i> zur Bildung von Statements als Alternative zum <i>Statement_Map</i> .....	63
Abbildung 19: UML Klassendiagramm des Moduls <i>Clinical Thesaurus</i> .....	64
Abbildung 20: UML Klassendiagramm zur Darstellung der Merkmalsebene und Wertebereiche für die Verwaltung von Dokumentationsmerkmalen in VokaDoks.....	65
Abbildung 21: UML Klassendiagramm zur Darstellung von unterschiedlichen Typen von Integritätsbedingungen in VokaDoks .....	67
Abbildung 22: Abbildung von InCoMe-Hauptelementen in ISO/IEC 11179 Ed.3.....	77

Abbildung 23: Ansicht zur Erfassung der freien Merkmalsarten [20] .....	77
Abbildung 24: Ansicht zur Erfassung der benötigten Stammdaten am Beispiel allgemeine Auswahlmöglichkeiten [20] .....	78
Abbildung 25: Erfassung von Merkmalsarten. Eine Patienten_ID darf höchstens sechs-stellig bis zum Maximalwert 888.888 sein. ....	79
Abbildung 26: Überarbeitung von Integritätsbedingungen von Merkmalsarten .....	80
Abbildung 27: Erfassung von Stammdaten durch manuelle Eingabe .....	81
Abbildung 28: Zuweisung von Stammdaten .....	81
Abbildung 29: Durchsuchen von standardisierten medizinischen Konzepten im Vokabularsystem. ...	82
Abbildung 30: Verfeinerung von Stammdaten durch Suchen in anderen medizinischen Vokabularen. ....	83
Abbildung 31: Übernahme von standardisierten medizinischen Begriffe und Möglichkeit der zusätzlichen Eingabe nicht standardisierter Begriffe .....	83
Abbildung 32: Übernahme von präkoordinierten Bezeichnungen von medizinischen Konzepten in die Stammdaten .....	84
Abbildung 33: Bereich der Anwendung: Startansicht zum Anlegen von Datenbank und Kontexten...	85
Abbildung 34: Ansicht zum Anlegen einer Datenbank.....	85
Abbildung 35: Ansicht zum Anlegen eines Kontexts .....	86
Abbildung 36: Hierarchie-Integritätsbedingung für die Datenbank Merkmalskatalog.....	87
Abbildung 37: Hierarchie-Integritätsbedingung für den Kontext Patient .....	87
Abbildung 38: Schlüssel-Integritätsbedingung: Ein Patient wird durch seine Patienten_ID eindeutig identifiziert. ....	88
Abbildung 39: Referenz-Integritätsbedingung .....	89
Abbildung 40: Semantische Integritätsbedingung.....	90
Abbildung 41: Semantische Integritätsbedingung.....	91
Abbildung 42: Sicherung des Entwurfsprojekts mit dem VokaDoks-Designer.....	91
Abbildung 43: Konfiguration des Designers zur Generierung der Datenbeschreibung und der Anwendungsbeschreibung für das Dokumentationssystem .....	92
Abbildung 44: Datenbeschreibung des modellierten Merkmalskatalogs als XML-Datei exportieren..	93
Abbildung 45: Datenbankskript zur Generierung der CDMS-Datenbank .....	94
Abbildung 46: Export der Konfigurationsdatei appl.xml für die CDMS-Dokumentationssystem-Applikation.....	95
Abbildung 47: Export der Webanwendungskonfigurationsdatei web.xml für die CDMS-Dokumentationssystem-Applikation .....	96
Abbildung 48: Modifizierungsbedarf der Anwendungskonfigurationsdatei (web.xml) .....	96
Abbildung 49: Anmeldung beim Tomcat-Manager .....	100
Abbildung 50: CDMS Anmeldung.....	101

Abbildung 51: Patientensuche nach den vordefinierten Suchkriterien .....	101
Abbildung 52: Ergebnisliste nach einer Suche nach Patienten und Möglichkeit einen neuen Patienten anzulegen.....	102
Abbildung 53: Form zum Anlegen eines neuen Patienten .....	102
Abbildung 54: Detailanzeige eines Patientendatensatzes.....	103
Abbildung 55: Visiten eines Patienten .....	103
Abbildung 56: Sozialdaten eines Patienten .....	103
Abbildung 57: Form mit Änderungsmöglichkeit eines Patientendatensatzes .....	104
Abbildung 58: Form zum Löschen eines Patientendatensatzes.....	104
Abbildung 59: Form zum Anlegen einer neuen Visite zu einem bestimmten Patienten .....	105
Abbildung 60: Form zum Anlegen von neuen Sozialdaten eines bestimmten Patienten .....	105
Abbildung 61: Beispiel einer Fehlermeldung bei Verletzung einer syntaktischen Integritätsbedingung .....	106
Abbildung 62: Beispiel einer Fehlermeldung bei Verletzung einer semantischen Integritätsbedingung .....	106
Abbildung 63: Consolidated Data Description Metamodel [31].....	121
Abbildung 64: Concept system metamodel region [31].....	122
Abbildung 65: Classification metamodel region [31] .....	122
Abbildung 66: Vereinfachtes TERMTrial Klassenmodell [Ableitung aus 47] .....	123
Abbildung 67: Klassenhierarchien in TERMTrial [Ableitung aus 47] .....	124
Abbildung 68: Vereinfachtes Klassendiagramm von InCoMe [Ableitung aus 20, 21].....	124
Abbildung 69: Modellelemente und deren Hierarchien in InCoMe [20, 21] .....	125
Abbildung 70: Vereinfachtes Datenmodell von ORCA [Ableitung aus 18, 48] .....	126
Abbildung 71: Klassendiagramm des Nursing Records Package [55].....	127
Abbildung 72: Klassendiagramm des Nursing Term Management Package [55] .....	128



## VERZEICHNIS DER TABELLEN

Tabelle 1: Erfüllungsstatus der Prüfkriterien vom ISO/IEC 11179 Ed.3 für eine terminologische Fundierung. ....	45
Tabelle 2: Zusammenfassung der Erfüllung der Prüfkriterien beim Bottom-Up-Ansatz.....	54
Tabelle 3: Erfüllungsstatus der Prüfkriterien für eine terminologische Fundierung .....	68
Tabelle 4: Zusammenfassung des Mapping von InCoMe-Modellelementen zu ISO/IEC 11179 Ed.3 Modellelemente .....	71
Tabelle 5: Zugangsdaten der VokaDoks-Vokabular-Datenbank.....	76
Tabelle 6: Übersetzung von Designer-Komponenten in SQL-Datenbankbeschreibung .....	93
Tabelle 7: Vergleich von VokaDoks mit den in der vorliegenden Arbeit analysierten Ansätzen anhand der Analyse Kriterien. ....	108



## VERZEICHNIS DER ABKÜRZUNGEN

<b>Abkürzung</b>	<b>Bedeutung</b>
AP	Arbeitspaket
BMBF	Bundesministerium für Bildung und Forschung
CDISC	Clinical Data Interchange Standards Consortium
CPR	Computer-based Patient Record
CRF	Case Report Form
CSS	Cascading Style Sheets
DDL	Data Definition Language
DIMDI	Deutsches Institut für Medizinische Dokumentation und Information
DIN	Deutsches Institut für Normung
ENRS	Electronic Nursing Record System
GPOH	(Deutsche) Gesellschaft für Pädiatrische Onkologie und Hämatologie
IBE	Institut für Medizinische Informationsverarbeitung, Biometrie und Epidemiologie
ICD-10-GM	Internationale Klassifikation von Krankheiten und verwandten Krankheitsproblemen – 10. Revision – German Modification
ICH	International Conference on Harmonisation
ICN	International Council of Nurses
ICNP	International Classification of Nursing Practice
ID	Identifikationsnummer
IDE	Integrated Development Environment
IEC	International Electrotechnical Commission
InCoMe	Integriertes System zur Computerunterstützten Entwicklung medizinischer Dokumentationssysteme
ISO	International Organization for Standardization
HTML	Hypertext Markup Language
J2EE	Java 2 Platform, Enterprise Edition
JDBC	Java Database Connectivity
JDOM	Java Document Object Model
JDK	Java Software Development Kit
JRE	Java Runtime Environment
JSP	Java Server Pages
JTC	Joint Project Committee
MedDRA	Medical Dictionary for Regulatory Activities
MDR	Metadata Repository
OID	Object Identifier

<b>Abkürzung</b>	<b>Bedeutung</b>
OOA	Object-Oriented Analysis
OOD	Object-Oriented Design
OPS	Operationen- und Prozedurenschlüssel
ORCA	Open Record for CAre
RUP	Rational Unified Process
SDE	Structured Data Entry
SDK	Software Development Kit
SNOMED-CT	Systematized Nomenclature of Medicine – Clinical Terms
SQL	Structured Query Language
TERMTrial	Terminology-based Remote-Data-Entry-System for clinical trials
TNM	Tumor Nodulus Metastase
UML	Unified Modeling Language
UMLS	Unified Medical Language System
V	Version
WHO	World Health Organization
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language Transformation

## A EINLEITUNG

---

In der medizinischen Wissenschaft muss jede wissenschaftliche Erkenntnis – empirisch oder theoretisch – dokumentiert und nachvollziehbar hergeleitet werden. Als einer der Kernfächer der medizinischen Informatik bezeichnet die medizinische Dokumentation die Methoden und Tätigkeiten des Sammelns, Erfassens, Erschließens, Ordnen, Aufbewahrens und gezielten Wiedergewinns medizinischer Informationen bzw. medizinischen Wissens zu spezifischen Frage- oder Aufgabenstellungen [17, 41, 42, 43]. Durch die medizinische Dokumentation ist heutzutage eine vernünftige und qualitativ hochwertige medizinische Behandlung und Versorgung der Patienten möglich. Sie dient auch der Gesundheitsberichterstattung, der Qualitätssicherung medizinischer Behandlungsprozesse im Gesundheitswesen, sowie eines effektiven Qualitätsmanagements und der Unterstützung der Forschung, indem klinische Information und Wissen „berechtigten Personen – und nur ihnen (Datenschutz) –, vollständig aber ohne Ballast, zum richtigen Zeitpunkt, am richtigen Ort und in der richtigen Form“ [41, 42] bereitgestellt werden. Zudem gilt eine standardisierte Dokumentation als Voraussetzung für die Qualität, die Kommunikation und den Austausch von Daten in der medizinischen Versorgung und zwischen Kooperationen von multizentrischen klinischen Studien [26, 47, 54].

Die Relevanz von Dokumentationssystemen – Systeme zur technischen und organisatorischen Unterstützung medizinischer Dokumentation – im Gesundheitswesen ist also heute nicht mehr zu demonstrieren. Man unterscheidet allerdings drei Bereiche für die Inhalte medizinischer Dokumentationen: klinische Informationen, medizinisches Wissen und Kerndaten des Gesundheitswesens. Dadurch ergibt sich eine Vielzahl möglicher Ziele einer medizinischen Dokumentation, so dass Folgendes feststehen darf: „Unterschiedliche inhaltliche Ziele – und in ihrem Gefolge eine unterschiedliche Frage- und Aufgabestellung – erfordern für ihre Umsetzung auch unterschiedliche Typen medizinischer Dokumentationssysteme.“ [43]. Es ist jedoch essenziell, dass diese Dokumentationssysteme bestimmte Anforderungen und Integritätsbedingungen erfüllen und ein bestimmtes konzeptuelles Design bzw. Modell unterstützen, um eine „durchgängige und übergreifende Sicherung der Qualität in medizinischen Dokumentationssystemen“ und somit eine bestmögliche Patientenversorgung zu erzielen [20].

Nach DIN 2342 ist die Terminologie definiert als die „Gesamtheit der Begriffe und ihrer Bezeichnungen und Definitionen in einem Fachgebiet“. Obwohl Terminologien Aufgaben von Klassifikationen im medizinischen Bereich kaum ersetzen werden können, werden sie heutzutage zunehmend relevant [45]. „Aufgrund der bei Klassifikationen fehlenden begrifflichen Ebene werden für verschiedene Domänen spezialisierte Terminologien bereitgestellt werden. Sie bieten mehr oder weniger maschinell verarbeitbares begriffliches und sprachliches Wissen.“ [28]. Zudem sollen durch Terminologie-Systeme Eindeutigkeit und Unterscheidbarkeit von Konzepten gewährleistet werden, einfacher Zugang zu detaillierten Texten und Datenbanken, sowie korrektes und vollständiges Schließen durch formale Darstellung in Ontologie ermöglicht werden, was Grundlage für

entscheidungsunterstützende Systeme ist [29, 67]. Somit repräsentieren Terminologien und Klassifikationen zwei prinzipiell zu unterscheidende Paradigmen medizinischer Dokumentation: einzelfallorientierte Dokumentation und einzelfallübergreifende Dokumentation [28]. Dabei stellen Terminologien Bezeichnungen und Definitionen von Dokumentationseinheiten und deren Eigenschaften bereit, durch Klassifikationen werden Wertebereiche von Merkmalskatalogen vereinheitlicht.

### A.1 GEGENSTAND UND MOTIVATION

Forschung wie Versorgung nutzen Dokumentationssysteme als Werkzeug zur Erfassung, Speicherung, Verarbeitung und Präsentation von medizinischen Daten. Derzeit ist die Situation durch eine Vielfalt an Angeboten von Dokumentationssystemen gekennzeichnet (Qualis®, Medical Office®, InCoMe, NEMESYS, ID DIACOS®, Agfa Healthcare, u.v.m.). So konkurrieren im Bereich elektronischer Akten rund 100 Anbieter von Praxisinformationssystemen, 20 Anbieter von Klinischen Arbeitsplatzsystemen für Krankenhäuser und 10 Anbieter von Gesundheitsakten für Patienten. „Wenn schon über die Art und den Umfang der durchzuführenden medizinischen Dokumentation bisher keine Einigkeit erzielt werden konnte, so wundert es nicht, dass Relevanz und Qualität des Datenmaterials, das vielen medizinischen Studien mit hohen Fallzahlen zugrunde liegt, wissenschaftlich exakten Kriterien nur selten standhalten“ [30]. So stellen Dokumentationssysteme für klinische Studien häufig Einzellösungen dar. Auch wenn Studienmanagementsysteme zur Planung und Durchführung von klinischen Studien eingesetzt werden, sind die jeweils umgesetzten Case Report Forms und Merkmalskataloge singuläre Zusammenstellungen für das einzelne Vorhaben.

Zudem wird die Datenstruktur in den Dokumentationssystemen von den jeweiligen Anbietern vorgegeben. In der Versorgung betrifft dies auch oftmals die Merkmalsdefinition [20, 47]. In der Versorgung muss man nicht selten mit einer unvollständigen medizinischen Dokumentation arbeiten, da Informationen bzw. detaillierte/präzise Datenbeschreibungen fehlen, was zu Zeitverlust und Fehlern bei Datenanalysen/-auswertungen führt [25].

Diese Mängel führen folglich oft zu minimaler Datenqualität. Die Standardisierung bzw. Harmonisierung und die Wiederverwendung der Merkmalsdefinitionen, die für andere Forschungsprojekte von Interesse sein können, könnten jedoch Arbeitseinsparungen und Qualitätssteigerungen von Studien bedeuten [47, 49].

Die oben genannten Missstände in der Planung empirischer Forschungsvorhaben werden daher als Barriere vernetzter Forschung und integrierter Versorgung in der Medizin betrachtet [26, 47, 55, 63, 65]. Die einrichtungsübergreifende Standardisierung bzw. Harmonisierung und die Vereinheitlichung bzw. Wiederverwendung von Merkmalskatalogen bzw. Merkmalsdefinitionen sind ein Weg um Planungsaufwand zu reduzieren, Datenqualität zu verbessern und neue Anwendungsszenarien zu eröffnen. Ein besonderer Schwerpunkt liegt dabei auf der Integration und Verwendung von kontrollierten Vokabularen zur Wissensorganisation und Anreicherung von Wertelisten bzw.

Ausprägungen von Merkmalen. Beispielhaft sei hier auf die Standardisierungsbemühungen des Clinical Data Interchange Standards Consortium (CDISC) sowie auf die internationale Initiative zur Einführung von dem Systematized Nomenclature of Medicine-Clinical Terms (SNOMED CT) hingewiesen [54]. Aktuelle Arbeiten im BMBF-geförderten Verbundvorhaben „Spezifikation und prototypische Implementierung eines Metadata Repository für die klinische und epidemiologische Forschung in Deutschland“ (Förderkennzeichen: 01EZ0936B) zeigen jedoch, dass für die Verknüpfung beider Lösungswege Konzepte und Modelle fehlen [51]. In seiner Habilitationsschrift hat sich Bürkle mit medizinischen Data Dictionaries (MDD) beschäftigt, um die Qualität und Effizienz medizinischer Informationssystemen zu verbessern bzw. die medizinische Dokumentation besser zu strukturieren [9, 66]. Dabei sollte ein MDD dazu dienen, ein kontrolliertes klinisches Vokabular in einem klinischen Informationssystem zu definieren, die semantischen Beziehungen zwischen den Vokabular-Begriffen abzubilden, und das Vokabular mit standardisierten externen Klassifikationen und Nomenklaturen zu verknüpfen. Obwohl dieser Ansatz mit einem Referenzmodell für MDD gerechtfertigt wurde, sind wir genauso wie Merzweiler der Meinung, dass die Lösung in der terminologischen Fundierung von Dokumentationssystemen liegt [26, 37, 47, 55]. Bei dieser Standardisierung von Dokumentationssystemen ergeben sich gleich mehrere Vorteile. Die Einheitlichkeit von Wertebereichen („Value Lists“) wird über standardisierte kontrollierte Vokabulare hergestellt. Zudem werden aber auch die Bezeichnungen und Definitionen von Dokumentationseinheiten und deren Eigenschaften (Merkmale, Item, Variable, etc.) von Terminologie-Systemen bereitgestellt. Aus Sicht der Datenhaltung betrifft die terminologische Fundierung sowohl die Objekte (z. B. Bezeichnung von Tabellen) als auch die Einträge. Damit wird die Harmonisierung von Merkmalskatalogen gleichsam im Nebenschluss über den Einsatz kontrollierter Vokabulare geschaffen. Insgesamt ergibt sich auch, dass verschiedene kooperierende klinische Studienzentren bei Vereinheitlichung ihrer Dokumentation durch terminologische Fundierung die Datenerfassung, den Datenvergleich und -austausch, sowie das Wiederauffinden von Daten erleichtern und dadurch eine höhere Datenqualität erzielen können.

Somit lassen sich auch insgesamt drei Hauptsichten der terminologischen Fundierung von Dokumentationssystemen unterscheiden, die bei der Lösung des Problems mit Hilfe eines Verfahrensmodells zu verbinden sind:

- a) Datenhaltung: Strukturen (Metadaten) zur Beschreibung von Daten, Einträge von Daten und Pflege dieser Daten,
- b) Dokumentation: Dokumentationseinheiten und Merkmale ggf. mit ihren Wertebereichen, die im Dokumentationssystem eingegeben werden sollen,
- c) Terminologie: Begriffe, Bezeichnungen, Konzepte von kontrollierten Vokabularen mit ihren Beziehungen zueinander, die für die Standardisierung der Dokumentation sorgen sollen.

Zusammengefasst liegt also der Schwerpunkt dieser Arbeit in der „terminologischen Fundierung von Dokumentationssystemen“. Über dieses Thema gibt es bisher nur wenige deutschsprachige

Veröffentlichungen. Die vorliegende Schrift gibt erstmals im deutschen Sprachraum einen ausführlichen Überblick über das Thema unter folgenden Gesichtspunkten:

- Zweck: Warum beschäftigt sich die vorliegende Arbeit mit Dokumentationssystemen und Terminologien?
- Semantik: Was bedeutet es ein Dokumentationssystem terminologisch zu fundieren?
- Nutzen und Methodik: Wieso ist die Thematik der terminologischen Fundierung von Dokumentationssystemen gerade wichtig und wie soll konkret diese Fundierung erfolgen bzw. wie können standardisierte medizinische kontrollierte Vokabulare als Grundlage für medizinische Dokumentationssysteme eingesetzt werden?

## A.2 PROBLEMSTELLUNG

Aus der Literatur konnten bereits relevante Beiträge zu der Fragestellung der Abbildung von Inhalten aus externen Standards (z.B. ICD-10, OPS, TNM, MedDRA, SNOMED-CT, etc. Siehe Grundlagen) auf die ISO/IEC CD2 11179-3 Information technology — Metadata registries (MDR) — Part 3: Registry metamodel and basic attributes vorgelegt werden [65]. Diese Beiträge bemängelten unter anderem die korrekte Verbindung bzw. Abbildung von Zusammenhängen zwischen diesen unterschiedlichen Vokabularen und Elementen der Dokumentation.

Dadurch, dass eine integrierte Gesundheitsversorgung zunehmend grenzüberschreitend stattfinden wird, prognostizierte Ingenerf, dass international vergleichbare Klassifikationen, die unterschiedliche medizinische Informationen beschreiben (z.B. Diagnosen, operative Maßnahmen, Behandlungsergebnisse, Medikation, usw.), über ein Mapping auf eine gemeinsame ausdrucksstarke Referenzterminologie realisiert werden sollten, damit deren Kompatibilität sichergestellt wird, und Behandlungsverfahren und somit Behandlungsergebnisse verbessert werden. SNOMED-CT ließe sich da als Referenzterminologie empfehlen [28, 56].

Da Dokumentationssysteme für klinische Studien heutzutage oft als sogenannte Remote-Data-Entry-Systems (RDE-Systeme) umgesetzt werden, bemängelte Merzweiler ihrerseits, dass es keine kommerziellen Systeme gibt, die einen umfassenden terminologischen Support anbieten, obwohl viele davon die Entwicklung von RDE-Systemen unterstützen.

In der klinischen Pflege und im medizinischen Umfeld im Allgemeinen wurden ebenfalls Mankos hinsichtlich Transparenz und Sichtbarkeit von Pflegedokumentationssystemen, Datenbeschreibung und Datenkonsistenz beobachtet, sowie die Wirkung von standardisierten Terminologien auf die Pflegedokumentation bzw. auf die Darstellung von Daten für eine strukturierte Dateneingabe evaluiert [5, 10, 11, 19, 26, 37, 55]. Thoroddsen stellte die Nutzung mehrerer standardisierter Pflgeterminologien in Europa fest, aber bemängelte zugleich den fehlenden allgemeinen Einsatz dieser Terminologien in der Pflege, was ein Hindernis für den Zugriff auf Pflegedaten darstellt bzw. die Wiederverwendung der Pflegedaten erschwert [34, 68].

Die Vielfalt an Angeboten von Dokumentationssystemen und der Mangel an einem übergreifenden terminologischen Support, der Relevanz- und Qualitätsmangel des Datenmaterials bei klinischen Studien, das Angebot von Einzellösungen von Dokumentationssystemen für klinische Studien sowie die jeweils umgesetzten elektronischen Case Report Forms (CRFs) und Merkmalskataloge als singuläre Zusammenstellungen für das einzelne Vorhaben, stellen alle zudem Missstände dar, die als Kernbarriere vernetzter Forschung und integrierter Versorgung gesehen werden.

Sicher ist, die Erbringung einer guten Dienstleistung in der Patientenversorgung, die effektive Kommunikation/Analyse von Patienten-/forschungsrelevanten Daten, sowie die Wiederverwendung solcher Daten, um zum Beispiel Behandlungsergebnisse messen zu können, hängen von der Qualität der verfügbaren Informationen ab, die Gesundheits-/Forschungseinrichtungen verwalten und miteinander austauschen. Somit wird es also deutlich, dass die Standardisierung von Dokumentationssystemen und dadurch die Qualität der Dokumentation tatsächlich den Einsatz und die Standardisierung der zugrunde liegenden Terminologie/kontrollierten Vokabularen impliziert [26, 34, 37, 47, 68]. Mehr Forschungsarbeiten und Auswertungsinformationen werden in Folge in der Standardisierung von Pflegedokumentation, medizinischer Dokumentation und Studiendokumentation dringend benötigt, um die Patientenversorgung deutlich und effizient zu verbessern, sowie den Versorgungsfortbestand sicherzustellen [26, 37].

### **A.3 ZIELSETZUNG**

Aus der Problemstellung lässt sich herleiten, dass es heutzutage ein zunehmendes Interesse an Kooperation und (Daten)Austausch zwischen medizinischen Einrichtungen sowie nationalen bzw. internationalen klinischen Studienzentren mit gleichen oder unterschiedlichen Vorhaben in unterschiedlichen medizinischen Bereichen gibt. Deswegen ist der wachsende Bedarf, Dokumentationssysteme für klinische Studien und medizinische/pflegerische Versorgung zu standardisieren, zu beobachten. Die terminologische Fundierung von Dokumentationssystemen kann dabei helfen eine bessere Qualität der medizinischen Dokumentation durch präzise Datenspezifikationen und -definitionen sowie durch eine bessere Datenstrukturierung zu erzielen.

Die Zielsetzung der vorliegenden Arbeit ist daher die Entwicklung eines im Gesundheitswesen benötigten integrierten Dokumentationsmodells zur Bereitstellung von Semantiken (Bezeichnungen und Definitionen) für Dokumentationseinheiten, Vereinheitlichung von Daten-Wertebereichen und Standardisierung von Dokumentationsmerkmalsarten durch den Einsatz von standardisierten Terminologien bzw. kontrollierten Vokabularen. Dementsprechend sollen die Ergebnisse dieser Arbeit letztendlich zu einer effizienteren, patientenübergreifenden und fachübergreifenden medizinischen Versorgung, sowie zur vernetzten Forschung und integrierten Versorgung beitragen.

Um also den Einsatz der terminologischen Fundierung von Dokumentationssystemen als Lösung zu den erläuterten Problemen aus der Problemstellung zu rechtfertigen, werden im Rahmen dieser Arbeit folgende Ziele im Einzelnen verfolgt:

1. Erarbeitung eines Modells zur Integration von Datenhaltung, Dokumentation und Terminologie in einem medizinischen Dokumentationssystem.
2. Beleg der Anwendbarkeit des integrierten Modells über eine Implementierung und prototypische Anwendung.

#### A.4 AUFGABENSTELLUNG

Aus den oben genannten Zielen dieser Arbeit leiten sich folgende Aufgabenstellungen ab:

1. Um ein Modell zur Integration von Datenhaltung, Dokumentation und Terminologien erarbeiten zu können, sollen:
  - a. eine Ist-Analyse bestehender Modelle zur terminologischen Fundierung von Dokumentationssystemen durchgeführt werden. Hierbei soll, wenn möglich, für jede Sicht des Modells (Datenhaltung, Dokumentation und Terminologie) ein Standardvertreter identifiziert und analysiert werden sowie auf seine Bezüge untersucht werden. Ein Bottom-Up- und ein Top-Down-Ansatz werden zur eigenen Modellfindung durchgeführt.
  - b. die benötigten Werkzeuge bzw. Software und Entwicklungsumgebung für die Erarbeitung des Modells und die spätere Softwareentwicklung beschrieben werden.
  - c. aus den in 1.a gefundenen Bezügen zwischen den drei Hauptsichten ein integriertes Modell zur terminologischen Fundierung von Dokumentationssystemen entwickelt werden. Hierbei leiten sich alle Anforderungen an das Modell von der Analyse bestehender Modelle in 1.a ab und müssen von dem zu entwickelnden Softwaresystem erfüllt werden. Das entworfene Modell wird mithilfe von UML Diagrammen beschrieben (Definitionsphase bzw. Objektorientiertes Analysemodell, OOA, mit dem Ziel ein allgemeines Modell zu erstellen).
  - d. das entworfene Modell in eine konkrete Softwarearchitektur überführt werden, die Informationen über technische Umsetzungsdetails enthält und direkt als Vorlage für die Implementierung des Dokumentationssystems in einer vorab ausgewählten Programmiersprache dienen wird (Objektorientiertes Design, OOD, weiterhin mithilfe von UML Diagrammen).
2. Um die Anwendbarkeit des aufgebauten integrierten Modells von Datenhaltung, Dokumentation und Terminologien belegen zu können, sollen:
  - a. die benötigten Werkzeuge bzw. die sachgerechte Entwicklungsumgebung, sowie das Vorgehensmodell für die Implementierung des Modells definiert werden.
  - b. die in Aufgabestellung 1.d entwickelte Software-Architektur des aufgebauten integrierten Modells implementiert bzw. als Web-Applikation in einer objektorientierten Programmiersprache (OOP) ausprogrammiert werden und ein relationales

Datenbanksystem für die Datenhaltung angebunden werden (relationale Datenbankmanagementsystem, RDMS).

- c. das System in eine ablauffähige Form gebracht und angewendet werden, sowie ein Test zur Validierung der prototypischen Anwendung des Modells durchgeführt werden (objektorientiertes Testen, OOT).

3. Diskussion und, falls möglich, Vergleich und Abgrenzung zu anderen Lösungen.

## **A.5 AUFBAU DER ARBEIT**

Die vorgelegte Arbeit gliedert sich in folgende Abschnitte:

### *Einleitung*

In Kapitel A wird auf die Thematik der Arbeit eingegangen sowie die Motivation erläutert, die diese Arbeit notwendig macht. Hierbei wird die Arbeit in einem fachlichen Kontext eingeordnet, damit der Gegenstand der Arbeit verständlicher und die erarbeitete Lösung des festgestellten Problems bzw. die Zielsetzung der vorliegenden Arbeit gerechtfertigt wird.

### *Aufarbeitung des Themas*

In Kapitel B werden zunächst einige für das weitere Verständnis dieser Arbeit notwendigen Begriffe definiert, der aktuelle Kenntnisstand aus der Literatur zusammengefasst, sowie die grundlegenden Konzepte dieser Arbeit eingeführt, und die in ihr verwendeten Methoden und Technologien erläutert. Diese Grundlagen betreffen medizinische Dokumentation und Dokumentationssysteme und medizinische kontrollierte Vokabulare.

### *Vorstellung bestehender Ansätze*

In Kapitel C werden das Vorgehen bzw. das Material und die Methoden zur Entwicklung eines eigenen Modells zur terminologischen Fundierung von Dokumentationssystemen erläutert. Hierbei widmet sich Kapitel C der Vorstellung und (Anforderungs-)Analyse bestehender Ansätze zur terminologischen Fundierung von Dokumentationssystemen (Top-Down- und Bottom-Up-Ansatz).

### *Bewertung bestehender Ansätze*

Kapitel D beschäftigt sich mit der Bewertung ausgewählter bestehender Ansätze zur terminologischen Fundierung von Dokumentationssystemen aus Kapitel C anhand festgelegter Analyse- und Qualitätskriterien. Anschließend folgt eine Zusammenfassung der Ergebnisse der Analyse.

### *Erstellung eines Modells zur terminologischen Fundierung von Dokumentationssystemen*

Der Entwurf eines integrierten Modells zur terminologischen Fundierung von Dokumentationssystemen baut auf die Ergebnisse der Analyse bestehender Ansätze zur terminologischen Fundierung von Dokumentationssystemen, sowie deren Anforderungsanalyse auf. Kapitel E setzt sich dementsprechend mit diesem Entwurf auseinander.

### *Prototypische Implementierung und Test*

Kapitel F gliedert sich in drei Haupt-Unterkapitel. Im Unterkapitel F1 wird das entworfene Modell in eine Architektur eines Anwendungssystems überführt. Hierbei wird die Architektur des entwickelten

Systems vorgestellt und die Implementierung ihrer zentralen Komponenten im Einzelnen beschrieben. Die in Kapitel B vorgestellten Konzepte und Technologien finden hier Anwendung. Nach der prototypischen Implementierung erfolgt ein Test zur Validierung des entwickelten Systems.

Das Unterkapitel F2 setzt sich mit den Ergebnissen aus der Implementierung des entwickelten Modells zur terminologischen Fundierung von Dokumentationssystemen auseinander und beschreibt den Ablauf, über den ein Anwender ein Dokumentationssystem spezifizieren kann. Das Unterkapitel F2 befasst sich ebenfalls mit den erforderlichen Systemeinstellungen zur einwandfreien Ausführung bzw. Installation der implementierten prototypischen Dokumentationsanwendung. Das Unterkapitel F3 beschreibt den Ablauf, über den ein Anwender das generierte Dokumentationssystem effektiv nutzen kann.

#### *Diskussion und Ausblick*

Während die in dieser Arbeit gestellten Aufgaben, ausgewählten Vorgehen und Ergebnisse am Stand der Literatur bzw. der Erkenntnisse im Kapitel G erörtert werden, befasst sich Kapitel H mit dem Ausblick der Arbeit.

#### *Zusammenfassung*

Kapitel I befasst sich zu guter Letzt mit der Zusammenfassung der vorliegenden Arbeit.

## **B GRUNDLAGEN**

---

### **B.1 GRUNDLAGEN DER MEDIZINISCHEN DOKUMENTATION**

#### ***B.1.1 Allgemeines***

Moderne Informations- und Kommunikationstechnologien unterstützen eine kooperative Patientenversorgung und vernetzte medizinische Forschung auf vielfältige Art und Weise. Aufgaben der medizinischen Informatik ist es dabei u.a. Methoden und Werkzeuge zur systematischen Planung der Datenverarbeitung in kollaborativen Umgebungen bereitzustellen [38, 54].

Der Abschnitt „Grundlage der medizinischen Informatik“ dient daher der Einführung in einen Teil der medizinischen Informatik, der für diese Arbeit Anwendung findet. Es werden hier die relevanten Grundlagen, verwendeten Werkzeuge, Methoden und Techniken dargelegt. Hierbei wird auf die Themen medizinische Dokumentation, Merkmalskataloge und medizinische kontrollierte Vokabulare näher eingegangen. Im Anschluss daran folgt eine Vorstellung der in dieser Arbeit verwendeten Technologien zur Entwicklung eines webbasierten Prototyps eines terminologisch fundierten Dokumentationssystems.

#### ***B.1.2 Medizinische Dokumentation und medizinische Dokumentationssysteme***

Eine übergreifende Analyse der Prozesse und der Anforderungen an medizinische Dokumentationssysteme ist für das Vorhaben dieser Arbeit unerlässlich, auch wenn die reine Planung und Implementierung eines Dokumentationssystems nicht Teil dieser Arbeit ist.

An dieser Stelle wird zunächst jeweils eine für diese Arbeit geeignete Definition der Begriffe medizinische Dokumentation und medizinisches Dokumentationssystem formuliert. Ihre Bedeutung sowie Anforderungen werden ebenfalls erläutert.

##### **B.1.2.1 Medizinische Dokumentation**

Die Bedeutung der medizinischen Dokumentation hat in den letzten Jahrzehnten deutlich zugenommen. Dies lässt sich durch eine vielfach differenziertere Diagnostik als früher, eine stets zunehmende Menge an Patientendaten, den hohen Kommunikationsbedarf zwischen Krankenhauspersonal, sowie den Bedarf an einer kooperativen Patientenversorgung erklären.

Unter medizinischer Dokumentation versteht man die Erfassung, Erschließung, Speicherung, Verarbeitung, Präsentation und Wiedergewinnung von medizinischen Informationen, sei es krankenhausspezifisch oder patientenbezogen, oder medizinischem Wissen (z.B. über eine Krankheit) [17, 41, 43].

Die medizinische Dokumentation spielt für die Medizin in der Patientenversorgung und in einer immer wachsenden vernetzten medizinischen/klinischen Forschung eine zentrale Rolle. Sie setzt in der Regel die Einwilligung jedes Patienten voraus, wird im Interesse jedes Patienten durchgeführt, dient als Kommunikationshilfe zwischen allen Versorgungsdienstleistern als Gedächtnisstütze „für geplante und durchgeführte Maßnahmen, hält die Untersuchungsergebnisse fest“ und wird zu

Abrechnungszwecken verwendet. Sie ist infolgedessen wichtig für eine vernünftige Behandlung der Patienten, für die Verwaltung und Gesundheitsberichterstattung, ist rechtlich vorgeschrieben und sichert die Qualität klinischer Forschungsergebnisse. Bei der medizinischen Dokumentation werden drei Hauptbereiche unterschieden [41]:

- Die patientenbezogene Dokumentation, deren angestrebte und heutige Form als „elektronische Patientenakte“ bezeichnet wird, umfasst medizinische Patientendaten. Krankengeschichte, klinische Basisdokumentation, Arztbrief und Befunddokumentation sind beispielsweise wichtige medizinische patientenbezogene Dokumentationen.
- Die Gesundheitsberichterstattung beinhaltet u.a. epidemiologische, sozialmedizinische und medizinstatistische Daten;
- Die Dokumentation des medizinischen Wissens ist patientenunabhängig und wird z.B. in Form von Fakten- und Wissensdatenbanken oder Literaturdatenbanken (zum Beispiel das Medical Literature Online (MEDLINE)) modelliert. Die Suche und das Wiederfinden von medizinischen Dokumenten werden hier unterstützt.

Zu den Zielen der patientenbezogenen medizinischen Dokumentation zählen u.a. „die wirkungsvolle Unterstützung der medizinischen Versorgung des einzelnen Patienten“, die Sicherung der medizinischen Versorgungsqualität von Patienten (durch Dokumentation von Behandlungsdaten), die Verallgemeinerung der Erfahrungen aus der Versorgung einzelner Patienten, sowie die Gewinnung neuer Erkenntnisse aus diesen Daten und die Unterstützung einer Recherche nach bestimmten Dokumenten [41, 43]. Als Kommunikationshilfe fördert zudem die medizinische Kommunikation die Übermittlung von Behandlungsdaten und den Austausch von Patienteninformationen und medizinischem Wissen zwischen Versorgungspersonal und Gesundheits- oder Forschungseinrichtungen.

Zudem sollte eine hochwertige medizinische Dokumentation folgende Qualitätsmaße besitzen [41]:

- Vollzähligkeit: alle dokumentierten Objekte müssen vorhanden und verfügbar sein.
- Vollständigkeit: alle Merkmale, die für jedes Objekt dokumentiert werden sollen, müssen vorhanden sein.
- Richtigkeit: die dokumentierten Merkmale stimmen mit der Realität überein (Validität, Gültigkeit).
- Beobachtungsgleichheit: für jedes einzelne Objekt werden z.B. dieselben Dokumentationsmethoden gleichartig angewendet.
- Strukturgleichheit: Übereinstimmung in der Verteilung aller Merkmale der beobachteten Objekte.
- Reproduzierbarkeit: Grad der Übereinstimmung der Dokumentation bei wiederholter Dokumentation desselben Objektes (Reliabilität, Zuverlässigkeit).

Typische medizinische Dokumentationen sind u.a. die Krankenakte, die Krankenaktenarchive, die klinische Basisdokumentation, die Befunddokumentation, die klinische Tumordokumentation, die

Dokumentation für das Qualitätsmanagement, klinische und epidemiologische Register, Dokumentation bei klinischen Studien und die Dokumentation in Krankenhausinformationssystemen. Für die Erreichung ihrer Ziele beschränkt sich die vorliegende Arbeit auf die patientenbezogene und forschungsbezogene Dokumentation, deren angestrebte Verbesserung durch das Ergebnis dieser Arbeit die Patientenversorgung unterstützen soll.

#### B.1.2.2 Medizinisches Dokumentationssystem

Ein medizinisches Dokumentationssystem stellt „alle für Diagnostik, Therapie und Pflege relevanten Informationen“ über einzelne Patienten in einer verständlichen Art und Weise bereit, ermöglicht die effiziente und effektive Durchführung von Dokumentationsaufgaben, die Verarbeitung von Patientensowie Administrationsdaten in der Versorgung und liefert vollständig, rechtzeitig und zuverlässig abrechnungsrelevante Informationen über die erbrachten Leistungen [41]. Somit wird die sinnvolle Nutzung der gesammelten Patienteninformationen, sowie eine schnelle, vollständige und auswertbare Dokumentation der Arzttätigkeit unterstützt und folglich das Qualitätsmanagement verbessert [17, 41, 43].

Um die Versorgung effizienter und angemessener zu machen, muss ein medizinisches Dokumentationssystem alle verfügbaren und relevanten Informationen über Patienten bereitstellen können. Die vorliegende Arbeit verfolgt die Modellierung eines horizontalen, direkten und standardisierten computerunterstützten Dokumentationssystems, da sich die angestrebte Dokumentation mit relativ wenigen Merkmalen bei relativ vielen Objekten beschäftigt, die Merkmale unmittelbar an den beobachteten Objekten (Patienten, Krankheiten, etc.) erfasst werden, und die Aufzeichnung der Merkmale einheitlich erfolgt (formatierte Dateneingabe, eCRF). Somit werden eine automatische inhaltliche Erschließung und die Auswertbarkeit der Dokumentation erzielt, und dadurch die semantische Interoperabilität zwischen unterschiedlichen Dokumentationssystemen gefördert. Unter Interoperabilität versteht man die Fähigkeit von zwei oder mehr Systemen Informationen auszutauschen und diese auch effektiv zu nutzen. Die semantische Interoperabilität ermöglicht zwei oder mehr Anwendungssystemen sich zu verstehen, indem sie ausgetauschte Informationen richtig interpretieren, verstehen und nutzen können [24, 54, 69]. Hier spielt das gemeinsame Wissen über die Bedeutung der verlässlich ausgetauschten Informationen eine zentrale Rolle. Im medizinischen Kontext fallen unterschiedliche Typen von Patientendaten (Textdaten, Labordaten, Röntgenbilder, Medikations- und Therapiedaten, etc.) an, die oft gemeinsam verarbeitet werden müssen, so dass die semantische Interoperabilität auch hier von großer Bedeutung ist. Sie ermöglicht den semantischen Austausch von medizinischen Daten für unterschiedliche Anwendungszwecke.

### ***B.1.3 Objekte, Merkmale, Merkmalskataloge***

Möchte man ein medizinisches Dokumentationssystem modellieren, muss man vorab die Gegenstände der Dokumentation, also die Liste aller relevanten zu dokumentierenden Informationen bzw. Merkmale, festlegen, sowie deren Bedeutung erläutern, da innerhalb einer Dokumentation ein Objekt durch seine Merkmale beschrieben wird [33]. Dabei wird in einem Dokumentationsprotokoll bzw. Merkmalskatalog festgelegt und beschrieben, welche Merkmale für welche Objekte mit welchen möglichen Ausprägungen dokumentiert werden.

Laut Leiner et al. stellt ein Objekt bzw. Gegenstand einen eindeutig identifizierbaren „Ausschnitt aus der wahrnehmbaren oder vorstellbaren Welt“ dar. Dabei besitzt jedes Objekt eine Menge von Eigenschaften, die es von anderen eindeutig unterscheidet (z.B. Frau Ngouongo Sylvie, geboren 30.12.1980, Universitätsklinikum München, Heuschnupfen) [43]. Gleichartige Objekte mit gemeinsamen Eigenschaften definieren einen Objekttyp (z.B. Patient, Krankheit, Krankenhaus). Eigenschaften von beobachteten Objekten werden innerhalb einer Dokumentation als Merkmale bezeichnet. Die beobachteten Objekte werden auch Dokumentationseinheit oder Merkmalsträger genannt.

Beispielsweise wird von der Dokumentationseinheit „Patient“ das Merkmal „Krankheit: Typhus abdominalis (A01.0)“ dokumentiert. Dabei bezeichnet die Merkmalsart „Krankheit“ die Eigenschaft des Merkmalsträgers „Patient“, die dokumentiert wird, und besitzt die Merkmalsausprägung „Typhus abdominalis (A01.0)“. Eine Merkmalsausprägung ist die zu einer Merkmalsart dokumentierte Wertemenge (z.B. [Heuschnupfen, Typhus abdominalis, Cholera]). In diesem Beispiel stammt die Merkmalsausprägung aus einer Menge von Klassen einer Krankheitsklassifikation (hier aus der Internationalen statistischen Klassifikation der Krankheiten und verwandter Gesundheitsprobleme, 10. Revision: ICD-10-GM).

Anhand dieses Beispiels lässt sich die Definition des Begriffs „Merkmal“ deutlicher formulieren. Ein Merkmal ist die Vereinigungsmenge einer Merkmalsart und der Menge der zu dieser Merkmalsart dokumentierten Merkmalsausprägungen. Merkmalsausprägungen werden innerhalb eines Dokumentationssystems auch als Datenobjekte (z.B. „Tuberkulose“, „weiblich“) und Merkmalsart auch als Datenobjekttypen (z.B. Krankheit, Geschlecht) bezeichnet.

In der klinischen Versorgung muss eine Basisdokumentation für jeden Patienten vorhanden sein. Eine Basisdokumentation in der Patientenversorgung besteht in der einheitlichen Erfassung einer festgelegten Anzahl von grundlegenden Merkmalen bei allen Patienten und enthält im Wesentlichen Diagnosen, operative Eingriffe und ausgewählte nicht-operative Prozeduren. Alle für die Basisdokumentation festgelegten Merkmale (Merkmalsarten und ihren entsprechenden Merkmalsausprägungen), sowie maßgebende Nomenklaturen und Kodierschlüssel werden in einem sogenannten Merkmalskatalog aufgeführt und beschrieben. Patientendokumentationsmerkmale in einer Basisdokumentation betreffen in der Regel Stammdaten (z.B. Vorname, Nachname, Geschlecht,

Geburtsdatum, Anschrift, Krankenkasse, etc.), Falldaten (z.B. Aufnahmegrund, Fachabteilung), Einweisungsdiagnose, Haupt- und Nebendiagnose, und Prozeduren.

#### **B.1.4 Medizinische kontrollierte Vokabulare**

Der Einsatz von Klassifikationen und Terminologien (zusammengefasst als kontrollierte Vokabulare) ist für eine Mehrfachverwendung von Daten zwingend erforderlich. Man spricht hier auch von semantischer Interoperabilität, da kontrollierte Vokabulare einen Austausch von Informationen auf der Bedeutungsebene unterstützen [24, 39]. Während die Standardisierung von Merkmalsdefinitionen für die empirische medizinische Forschung erst am Anfang steht, wird bereits eine Vielzahl von Terminologien und Klassifikationen insbesondere in der medizinischen Versorgung und in der Forschung evaluiert, eingesetzt oder qualitativ verbessert [10, 19, 26, 28, 29, 36, 56, 58, 61]. Beispielhaft sei auf die Internationale statistische Klassifikation von Krankheiten und verwandter Gesundheitsprobleme (ICD-10, s. <http://www.dimdi.de/>), die TNM-Klassifikation (s. <http://www.uicc.org>), das Medical Dictionary for Regulatory Activities (MedDRA, s. <http://meddrasso.com/>), sowie Systematized Nomenclature of Medicine Clinical Terms (SNOMED-CT, s. <http://www.ihtsdo.org/>) hingewiesen. Angesichts des immer mehr wachsenden Interesses und der vielfältigen Einsatzmöglichkeiten von kontrollierten Vokabularen wurden ebenfalls bereits Anforderungen bzw. Wünsche an diese gestellt, um medizinische Anwendungen wie medizinische Dokumentationssysteme optimal zu gestalten bzw. zu unterstützen [11].

Oft werden Begriffe oder Klassen eines kontrollierten Vokabulars Konzepte genannt. Ein Konzept bezeichnet dabei einen abgrenzenden klinischen Inhalt, der eindeutig identifiziert werden kann – durch eine Concept ID oder einen Kode) –, eine eindeutige Bezeichnung trägt, eine oder mehrere Beziehungen zu anderen Konzepten innerhalb des Vokabulars hat und ggf. synonyme Bezeichnungen besitzt [11].

##### **B.1.4.1 Medizinische Terminologien**

„Eine Terminologie ist der Gesamtbestand der Begriffe (repräsentiert durch ihre Definitionen) und ihrer Benennungen in einem Fachgebiet; sie wird auch als Fachwortschatz bezeichnet.“ [43]. Laut dem deutschen Terminologie-Portal ist eine effiziente fachsprachliche Kommunikation ohne korrekte Verwendung von Fachwörtern nicht möglich. Folglich beschäftigt sich die Terminologie-Arbeit mit der Erarbeitung, Bearbeitung, Speicherung und Nutzung von Fachwörtern (Deutsches Institut für Terminologie e.V., s. <http://www.termportal.de/>). Im Folgenden sollen medizinische Terminologien vorgestellt werden, die im Rahmen dieser Arbeit von Bedeutung sind.

###### **B.1.4.1.1 MedDRA – Medical Dictionary for Regulatory Activities**

Als internationale medizinische Terminologie dient das Medizinische Wörterbuch für Aktivitäten im Rahmen der Arzneimittelzulassung (MedDRA) der Klassifikation von unerwünschten Arzneimittelwirkungen in Verbindung mit dem Konsum von biopharmazeutischen und anderen

medizinischen Produkten (s. <http://meddramsso.com/>). Sie wurde unter der Federführung der International Conference on Harmonisation (ICH) entwickelt. MedDRA umfasst standardisierte medizinische Begriffe, die in dem Arzneimittelentwicklungsprozess verwendet werden, und ist von besonderer Bedeutung für die elektronische Übermittlung von unerwünschten Ereignissen von Arzneimittel (Nebenwirkungen) sowohl in der Prä- als auch Post-Marketing Phase von Fertigarzneimitteln. Auf seiner untersten Ebene umfasst MedDRA über 66.000 Bezeichnungen (so genannte Terme), die über verschiedene Stufen und Pfade in 26 systematischen Klassifikationen in der obersten Ebene zusammengeführt werden. Diese Klassen in der obersten Ebene bilden das Organklassensystem (Organ Class System, SOC). Jede Ebene ist durch eine bestimmte Granularität charakterisiert, wobei die unterste Ebene (LLT, Lowest Level Term: Begriffe niedrigster Ebene) maximale Spezifität liefert. Jeder Begriff bzw. Term wird über einen 8-stelligen numerischen Kode verschlüsselt. Für eine Datenbankabfrage steht das Tool „Standardised MedDRA Query“ (SMQ, standardisierte MedDRA-Abfragen) zur Verfügung. Die Struktur von MedDRA erlaubt die Vergleichbarkeit unerwünschter Arzneimittelwirkungen, ihre Addition und die Zusammenfassung entlang definierter Pfade.

#### ***B.1.4.1.2 SNOMED-CT – Systematized Nomenclature of Medicine – Clinical Terms***

Die systematisierte Nomenklatur der Human- und Veterinärmedizin ist die wichtigste allgemeine Nomenklatur der Medizin. SNOMED-CT ist eine umfassende, konzept-basierte und mehrachsige medizinische Terminologie, die medizinische Objekte bzw. Begriffe über Bezeichnungen aus 7 semantischen Hauptachsen (T – Topographie, M – Morphologie, E – Ätiologie, F – Funktionsstörung, D – Diagnose, P – Prozeduren, J – berufliche Belastungen), sowie deren Beziehungen untereinander beschreibt. Ihr Primärziel ist durch die Unterstützung der klinischen Dokumentation die Patientenversorgung zu verbessern [6]. Durch ihre bedeutende Ausdrucksfähigkeit kann sie für medizinische Dokumentationen, wie die elektronische Gesundheitsakte, eingesetzt werden. Durch die detaillierte Beschreibung von medizinischen Begriffen in mehreren Achsen eignet sich SNOMED-CT zudem für das Wiederfinden möglichst vieler Dokumentationsobjekte mit einem bestimmten Merkmal und die Auswertung klinischer Daten. Die Anwendung von SNOMED-CT erfolgt vorwiegend über die Festlegung der Beziehungen zwischen verschiedenen Achsen bei Anwendung, die sogenannte Postkoordination [60, 71]. Die Präkoordination ist die Erzeugung zusammengesetzter medizinischer Begriffe, deren Beziehungen in der Definitionsphase vor Anwendung festgelegt werden.

Beispiel für Postkoordination (nach Leiner): Ein Schiffskoch "J53150" wird mit den Symptomen Fieber "F03003", Schüttelfrost "F03260" und Diarrhoe "F62400" als Notfall in ein Krankenhaus aufgenommen "P00300". Dort wird eine akute Entzündung "M41000" des Magens "T63000" und des Duodenums "T64300", hervorgerufen durch Salmonella cholerae-suis "E16010", diagnostiziert, und die Diagnose Gastroenteritis paratyphosa "D01550" gestellt.

#### B.1.4.2 Medizinische Klassifikationen

"Klassifikationen sind Ordnungssysteme, die auf dem Prinzip der Klassenbildung beruhen. In den Klassen werden Begriffe zusammengefasst, die in mindestens einem klassenbildenden Merkmal übereinstimmen. Klassifikationen sind dafür gedacht, die Menge aller in einer Klasse zusammengefassten Gegenstände zu beschreiben.“ [43] Mit Klassifikationen können folglich Antworten auf statistische Fragestellungen gegeben werden; beispielsweise unterstützen sie, wenn man eine Statistik über die Häufigkeiten von Diagnosen, Befunde oder operative Eingriffe in einer Klinikabteilung oder Praxis benötigt. Außerdem ermöglichen sie patientenübergreifende Auswertungen und die Suche nach gleichartigen Dokumentationsobjekten. Klassen einer Klassifikation sollten sich nicht überschneiden und das jeweilige Fachgebiet vollständig überdecken. Jede Klasse in einer monohierarchischen Klassifikation hat höchstens eine übergeordnete Klasse (außer der Wurzel der Hierarchie) und jede untergeordnete Klasse ist entweder eine Spezialisierung oder ein Teil der übergeordneten Klasse (Generalisierung). Eine Klasse in einer Polyhierarchie kann hingegen mehrere übergeordnete Klassen haben [41, 43].

Im Folgenden sollen medizinische Klassifikationen vorgestellt werden, die im Rahmen dieser Arbeit von Bedeutung sind.

##### ***B.1.4.2.1 ICD-10-GM 2010 – Internationale statistische Klassifikation der Krankheiten und verwandter Gesundheitsprobleme, 10. Revision, German Modification, Version 2010***

Die ICD-10 ist eine monohierarchische Klassifikation, die Krankheiten primär nach Ätiologie (Krankheitsursache) und Topographie einteilt. Für jede Krankheit gibt es in der ICD-10 genau eine (passende) Klasse und einen Kode. Die ICD-10, die ursprünglich als internationales Verzeichnis der Todesursachen verwendet wurde, ist heute die wichtigste, weltweit anerkannte Diagnoseklassifikation in der medizinischen Versorgung. Die deutsche Variante (German Modification) wird durch das Deutsche Institut für Medizinische Dokumentation und Information (DIMDI) jährlich herausgegeben und gesetzlich in Deutschland zur Kodierung von Krankheiten und zu Abrechnungszwecken in der medizinischen Versorgung eingesetzt [64].

ICD-10-GM 2010 gliedert sich hierarchisch in 22 Krankheitskapitel (z.B. Kapitel I „Bestimmte infektiöse und parasitäre Krankheiten“), 241 Krankheitsgruppen (z.B. A00-A09 „Infektiöse Darmkrankheiten“), 1.709 Krankheitskategorien (z.B. A01.- „Typhus abdominalis und Paratyphus“), und 13.315 Krankheitssubkategorien, auch endständige Klassen oder Klassen der tiefst möglichen Hierarchieebene genannt (z.B. A01.0 „Typhus abdominalis“) [Zahlen aus ICD-10-GM 2010, s. <http://www.dimdi.de/static/de/klassi/icd-10-gm/index.htm>].

##### ***B.1.4.2.2 OPS - Operationen- und Prozedurenschlüssel***

OPS ist eine monohierarchische Klassifikation, die medizinische Maßnahmen primär nach Art des Eingriffs (diagnostisch/therapeutisch, operativ/nicht operativ) und nach Lokalisation des Eingriffs

einteilt. OPS basiert auf der International Classification of Procedures in Medicine (ICPM) der WHO von 1978 – übernommen und ausgebaut wurden die Kapitel 1 (Diagnostische Maßnahmen), 3 (Bildgebende Diagnostik), 5 (Operationen), 6 (Medikamente), 8 (Nichtoperative therapeutische Maßnahmen) und 9 (Ergänzende Maßnahmen). OPS wird ebenfalls durch DIMDI jährlich herausgegeben und gesetzlich in Deutschland zur Kodierung der stationären oder ambulanten operativen Maßnahmen und anderer medizinischen Prozeduren und zu Abrechnungszwecken eingesetzt [s. <http://www.dimdi.de/static/de/klassi/ops/index.htm>].

#### ***B.1.4.2.3 TNM – Tumor-Nodule-Metastasis Staging***

TNM ist eine Klassifikation, die der Stadieneinteilung von malignen Tumoren (Krebserkrankungen) dient und basiert auf statistischen Untersuchungen, die Aussagen über das voraussichtliche Verhalten von Tumorerkrankungen zulassen. Die Einstufung einer Tumorerkrankung in die einzelnen Kategorien des TNM-Systems erlaubt daher prognostische Aussagen und bestimmt häufig auch die weitere Therapie [62]. Die TNM-Klassifikation von bösartigen Tumoren ist ein etablierter Standard in der Onkologie. T steht für Tumor und beschreibt die Größe, die Ausdehnung und das Verhalten des Primärtumors. N steht für Nodus (d.h. Knoten) und bezeichnet die angrenzenden Lymphknoten, die von Absiedlungen des Tumors befallen sind. M kennzeichnet das Fehlen oder Vorhandensein von (Fern-)Metastasen, also entfernte Absiedlungen und Tochtergeschwülste in anderen Organen. Jedem Buchstaben folgt eine Zahl, die das jeweilige T, N und M-Stadium wiedergibt. Das so genannte „Staging“ (Krankheitsstadium) ist folglich für den Therapieplan entscheidend, da es kennzeichnet, wie stark sich die Krankheit bereits entwickelt hat. T2N0M0 beschreibt beispielsweise ein Melanom mit einer Dicke von 1-2mm ohne Metastasen in Lymphknoten und anderen Organen.

#### ***B.1.4.2.4 ICNP – International Classification of Nursing Practice***

Die vom International Council of Nurses (ICN) entwickelte internationale Klassifikation für die Pflegepraxis (ICNP, s. <http://www.icnp.info/>) ist eine mehrachsige kombinatorische Klassifikation und dient der Erstellung und Darstellung von Pflegeprozessen: Befunde, Diagnosen, Pflegemaßnahmen und Pflegeausgänge [36, 37]. Die ICNP Klassifikation ermöglicht dem Pflegepersonal seine Aktivitäten standardisiert zu formulieren und die Dokumentation der Pflegepraxis zu verbessern. Ein Teil von ICNP-Konzepten wird für die Präkoordination angewendet. Durch seine Multi-Axialität können nämlich Begriffe aus verschiedenen Achsen kombiniert und so komplexe Aussagen aus der Pflegepraxis formuliert werden.

#### ***B.1.5 Abschnittszusammenfassung***

Eine unzureichende, unvollständige und fehlerhafte medizinische Dokumentation gefährdet sowohl die Patientenversorgung, -sicherheit und -wohlbefinden als auch den Versorgungsfortbestand [37]. Die medizinische Dokumentation spielt dementsprechend für die Medizin in der Patientenversorgung und in einer ständig wachsenden vernetzten medizinischen/klinischen Forschung eine zentrale Rolle. Sie

stellt die Qualität von klinischen und medizinischen Daten sicher und ermöglicht die effektive und effiziente Umsetzung von Dokumentationsanforderungen, die Verarbeitung von stets zunehmenden Mengen an Patienten- und administrativen Daten, die Erstellung von differenzierteren Diagnostiken, sowie die Deckung des Bedarfs einer kooperativen Patientenversorgung und des hohen Kommunikationsbedarfs zwischen Krankenhauspersonal in der medizinischen Versorgung. Es ist daher nicht verwunderlich, dass die Relevanz der medizinischen Dokumentation in den letzten Jahrzehnten deutlich zugenommen hat.

Medizinische Dokumentationssysteme nutzen infolgedessen medizinische Dokumentation, um präzise, relevante und zuverlässige Patienteninformationen zeitgerecht zur Verfügung zu stellen und werden in der medizinischen Versorgung als Werkzeuge für die Sammlung, Erfassung, Speicherung, Verarbeitung, Darstellung und Pflege von medizinischen Patienten- und administrativen Daten verwendet.

Terminologien bzw. kontrollierte Vokabulare setzen Konsistenz und Granularität zur Beschreibung von medizinischen Konzepten und Fakten voraus. Die Qualität der beschriebenen Daten lässt sich somit erhöhen; das Erschließen, die Wiederverwendung, sowie Wiederfindung von Daten zum Auswertungszweck werden unterstützt.

Der zunehmende Bedarf Dokumentationssysteme terminologisch zu fundieren bedeutet, diesen Dokumentationssystemen einen Terminologie-Support zu verleihen. Die Einbindung von Terminologien und Klassifikationen in ein Dokumentationssystem ist für das heutige Management im Gesundheitswesen eine wesentliche Forderung [11, 26, 37, 47, 55, 56]. Forschungsarbeiten in der Auswertung von standardisierten Klassifikationen und Terminologien für die medizinische/klinische Dokumentation zeigten bereits, dass die Standardisierung eines Dokumentationssystems durch die Nutzung von standardisierten medizinischen Klassifikationen, Terminologien oder Ontologie in der Tat nicht nur die Konsistenz und die Qualität der Dokumentation der Patienten- und klinischen Daten sowohl in der medizinischen Versorgung als auch in der Pflege verbessert [11, 19, 26, 28, 29, 34, 37], sondern die semantische Interoperabilität auch fördert, indem ein einheitliches Verständnis zur Gesundheit, Krankheit, Diagnostik, Therapie, etc. verliehen wird [69].

Die terminologische Fundierung von Dokumentationssystemen ist gerade deshalb so wichtig, da kontrollierte Vokabulare noch mehr Vorteile mit sich bringen: Die Nutzung unterschiedlicher Terminologien in der Pflegedokumentation ermöglicht zum Beispiel die Wiederverwendung der Daten in unterschiedlichen medizinischen Anwendungsebenen, wie Pflegeplanung, Entscheidungsfindung in der Patientenversorgung, sowie klinische Forschung, Management im Gesundheitswesen, Gesundheitsplanung und staatliche Berichterstattung. Die Standardisierung von Diagnosen (z.B. ICD-10) und die Standardisierung von medizinischen Maßnahmen (z.B. OPS) zum Dokumentationszweck in der medizinischen Versorgung (z.B. in der elektronischen Patientenakte) ermöglichen die internationale Sichtbarkeit und Messbarkeit/Quantifizierbarkeit/Vergleichbarkeit der Pflege bzw. der Versorgung [26, 27]. Die Verknüpfung mehrerer Vokabulare miteinander wird ebenfalls von Häyrinen

et al. empfohlen, um die Versorgung von Patienten elektronisch nachzuverfolgen. So können zum Beispiel ICD-10-kodierte Diagnosen für die Identifikation von unerwünschten Arzneimittelereignissen (UAE) in Routinedaten genutzt werden [64], um die UAE dann mit MedDRA kodiert für Zwecke der Verbesserung der Arzneimittelsicherheit weiterzuverarbeiten [2].

Auf Merkmalebene (z.B. für Merkmalskataloge für klinische Register) eignen sich Terminologien und Klassifikationen zudem sehr gut für die Festlegung von Wertebereichen von Merkmalen mit aufzählender Charakteristik (z. B. Patientengeschlecht, Aufnahmeart, Diagnosentyp), aber auch für die Festlegung von Wertebereichen von Merkmalen mit komplexer Charakteristik (z. B. Diagnosen, Prozeduren, Komplikationen, unerwünschte Arzneimittelwirkungen). Außerdem unterstützt die Standardisierung von medizinischen Dokumentationssystemen durch kontrollierte Vokabulare die Vergleichbarkeit bzw. Harmonisierung von Datenobjekten, sowie Entscheidungsfindungen im klinischen Umfeld [1, 26, 43, 44, 45].

Und nicht zuletzt, ein standardisiertes und effizientes medizinisches bzw. pflegerisches Dokumentationssystem bewirkt einen eher positiven Effekt auf die Akzeptanz von Computergestützten Systemen im Gesundheitswesen [37].

In Anbetracht des beobachteten Bedarfs an weiteren und vertieften Auswertungsstudien hinsichtlich der Nutzung und der Wirkung von kontrollierten Vokabularen auf die elektronische medizinische/Pflege- Dokumentation [26, 27], zielt die vorliegende Arbeit darauf ab seinen Beitrag zu leisten, indem Qualitätskriterien zur terminologischen Fundierung von Dokumentationssystemen erarbeitet und bestehende Ansätze analysiert werden, und ein Dokumentationsmodell mit terminologischer Fundierung erarbeitet wird. Die Implementierung eines solchen Dokumentationsmodells soll die Erstergebnisse dieser Arbeit rechtfertigen.

## **B.2 TECHNISCHE GRUNDLAGEN**

In diesem Abschnitt werden Standards und objektorientierte Modellierungs- und Entwicklungstechniken vorgestellt, die bei der Entwicklung eines prototypischen webbasierten terminologisch fundierten Dokumentationssystems Einsatz finden.

Die Wahl ist auf die objektorientierte Modellierung gefallen, da sich ihre Eigenschaften zur Datenabstraktion, -vererbung, und Polymorphismus, sowie die Fähigkeit von Objekten, die Komplexität der realen Welt zu modellieren, für die Entwicklung komplexer Anwendungen gut eignen [14]. Zum besseren Verständnis wird als Beispiel das von Goertzen implementierte integrierte System zur computerunterstützten Entwicklung von medizinischen Dokumentationssystemen namens InCoMe genommen. InCoMe findet auch Einsatz in dem Abschnitt Material und Methoden dieser Arbeit.

### **1. Java**

Java ist eine objektorientierte Programmiersprache, die 1991 von Sun Microsystems entwickelt wurde. Hinter Java EE (Java Plattform, Enterprise Edition) verbirgt sich die Spezifikation einer Softwarearchitektur für die transaktionsbasierte Ausführung von in Java programmierten

Anwendungen, insbesondere Web-Anwendungen. (s. <http://www.oracle.com/us/technologies/java/overview/index.html>). Interessant bei Java ist die Fähigkeit einen ausführbaren Code plattformunabhängig auszuführen. Goertzen griff zum Beispiel zur Implementierung von InCoMe, ein integriertes System zur computerunterstützten Entwicklung von medizinischen Dokumentationssystemen, auf Java-Bibliotheken wie Java Foundation Classes (JFC) und Swing zurück, um die graphische Benutzeroberfläche zu erstellen. Goertzen griff ebenfalls auf Java EE-Komponenten zurück, welche eine spezielle Infrastruktur, sogenannte Java EE Applikationsserver, erfordern, um J2EE-Spezifikationen zu unterstützen und herstellerepezifisch zu implementieren [20, 21]. So wurde das auf InCoMe basierte Dokumentationssystem, das Clinical Data Management System (CDMS), als Web-Anwendung implementiert.

## 2. UML – Unified Modeling Language

Die Unified Modeling Language (UML) ist eine standardisierte Notationssprache zur Beschreibung der objektorientierten Modellierung von Strukturen oder (Software-)Systemen, die von der Object Management Group (OMG, [www.omg.org](http://www.omg.org)) gefördert wird [35]. UML kombiniert drei führende objektorientierte Methoden (Booch, OMT, und OOSE) und stellt somit Systemarchitekten, Software-Ingenieuren, und Softwareentwicklern Werkzeuge zur Verfügung, softwarebasierte Systeme zu analysieren, zu entwerfen und zu implementieren. UML ermöglicht zudem die Modellierung von Geschäftsprozessen bzw. -abläufen und spielt eine wichtige Rolle in der Umsetzung der modellgetriebenen Architektur [57]. Zur Beschreibung des von Goertzen implementierten InCoMe-Modells wurden zum Beispiel Diagrammtypen wie Anwendungsfalldiagramme und Klassendiagramme erstellt. Somit wurden klar, welche Objekte der realen Welt in welcher Struktur und zu welchem Zweck benötigt werden und vor allem in welchen Beziehungen sie mit anderen Objekten der Struktur stehen. Beispiel: Nurse A gibt Patientendaten im InCoMe-Dokumentationssystem ein, damit Arzt B sie später auswertet. Nurse A ist dabei eine Mitarbeiterin vom Arzt B.

## 3. Model Driven Architecture (MDA)

Die modellgetriebene Architektur (Model Driven Architektur, MDA) bezeichnet einen modellgetriebenen Softwareentwicklungsansatz mit Architekturzentrierung, der auf verschiedenen Methoden und Konzepten, wie z.B. UML formale Spezifikationen, komponentenbasierte Softwareentwicklung, Architekturmuster, Analyse und Design, und Modelltransformationen, basiert. MDA beruht auf einer klaren Trennung von Prozessen (Funktionalität und Technik) der Softwareentwicklung. Projektspezifische Aspekte (fachliche Klassen, Geschäftsfunktionen) werden nämlich von technischen Aspekte (z.B. Plattformdefinitionen) getrennt [57].

## 4. Vorgehensmodell

Die objektorientierte Analyse und das objektorientierte Design (OOA/OOD) nach Booch [7] sind zwei wichtige Entwicklungsphasen eines Softwaresystems und werden als Vorgehensmodell angewandt.

Die objektorientierte Analyse dient der Definition und Analyse des Softwaresystems und das objektorientierte Design dient dessen Modellierung und Architektur.

Mit Hilfe des OOA/D konnte zum Beispiel Goertzen eine Fortschrittskontrolle des zu entwickelten Entwurfsmoduls (CDMS-Designer) durch Meilensteine durchführen und dadurch die Planung des Entwicklungsprozesses der CDMS-Anwendung wesentlich erleichtern. UML ermöglichte dabei die standardisierte Dokumentation beider Entwicklungsphasen.

##### 5. Architekturmuster (Design Patterns)

Architekturmuster ermöglichen komplexe Sachverhalte der Softwareentwicklung schnell und einfach darzustellen. Der Model-View-Controller (MVC) ist ein Architekturmuster, das die Softwareentwicklung in Modellierungsebenen strukturiert: Das Datenmodell (Model) beschreibt alle Daten, die durch das Softwaresystem verwaltet werden (sogenannte Fachklassen), und dient der Datenhaltung und -verarbeitung. Die Darstellungsschicht (View) dient der Präsentation benötigter Daten aus dem Datenmodell, sowie der Interaktion mit dem Benutzer. Die Steuerungsschicht (Controller) überwacht Benutzeraktionen auf der Präsentationsebene, in dem sie diese auswertet und zur Beantwortung von Benutzerabfragen auf die Daten zugreift und entsprechende Antwortsichten (Views) mit den benötigten Daten aufruft. Die Steuerungsschicht steuert somit die Geschäftslogik des Softwaresystems. Das Model-View-Controller-Architekturmuster ermöglicht ein flexibles Softwaredesign und eine flexible Softwareentwicklung, sowie die Wiederverwendung von einzelnen Softwarekomponenten, in dem Datenhaltung, Präsentationslogik und Geschäftslogik voneinander getrennt implementiert werden [8, 22, 23, 58]. Goertzen strukturierte zum Beispiel sein Java basiertes InCoMe-Projekt zur Entwicklung des CDMS-Designers gemäß des Architekturmusters MVC und mit folgenden Modellierungsebenen/Paketen: [entities] und [logical] zur Definition der Fachklassen (Model), [editors] und [gui] zur Implementierung sämtlicher graphischen Oberflächen des Designers (Views) und [designer] mit Klassen zur Steuerung aller Geschäftsprozesse bzw. Anfragen zwischen Model und Views. Die im Paket [gui] implementierten Klassen wurden so allgemein gehalten, dass sie wiederverwendet bzw. erweitert werden können.

##### 6. Relationales Datenbankmanagementsystem

Relationale Datenbanken dienen der Modellierung von Datenstrukturen und der Speicherung bzw. Verwaltung von Daten in einer Datenbank. Sie basieren auf dem relationalen Datenmodell von Edgar Codd in Form von zweidimensionalen Tabellen. Zum Einsatz kommen beispielhaft MySQL-, PostgreSQL, Oracle- oder Microsoft Access-Datenbanken. Relationale Datenbanken werden in einem relationalen Datenbankmanagementsystem (RDBMS) verwaltet. Für diese Arbeit wurde die Wahl auf PostgreSQL und MySQL gerichtet, da es gegenüber kommerziellen und damit kostenpflichtigen Möglichkeiten der Datenhaltung eine frei verfügbare und verwendbare Software unter Open-Source-Lizenz ist. Zum Abfragen und Manipulieren von Daten einer relationalen Datenbank wird meistens auf die Datenbanksprache SQL (Structured Query Language) zurückgegriffen [15]. Goertzen ermöglichte in seinem Designer die Erzeugung sowohl von MySQL- als auch von Oracle-

Datenbankskripten für die persistente Speicherung der in CDMS einzugebenden Daten durch Einbindung von Datenbank-Plugins. Die relationale Datenbank von Oracle ist jedoch kostenpflichtig.



## C MATERIAL UND METHODEN

---

### C.1 VORGEHEN

Top-Down- und Bottom-Up-Ansätze sind zwei gegenteilige Software-Entwicklungsprozesse in der Informatik, die auch häufig zur Systemanalyse bzw. -synthese eingesetzt werden. In einem Top-Down-Ansatz wird ein Überblick über das System verschafft. Dabei werden erste Subsysteme so spezifiziert, jedoch nicht ausführlich dargelegt, dass das gesamte System nur auf seine Basiselemente reduziert wird. In einem Bottom-Up-Ansatz hingegen werden zuerst einzelne Basiselemente des Systems ausführlich spezifiziert und dann miteinander verlinkt, um größere Subsysteme zu gestalten. Diese wiederum werden miteinander verlinkt bis ein vollständiges neues System gebildet wird [32].

Aufgrund der Angemessenheit dieser beiden Verfahren zur Beantwortung der im Rahmen dieser Arbeit erhobenen Fragestellung, verfolgt die vorliegende Arbeit sowohl einen Top-Down- als auch einen Bottom-Up-Ansatz.

In dem Bottom-Up-Ansatz wurde die Literatur auf Best Practices in der Entwicklung von Terminologie-basierten Dokumentationssystemen in der medizinischen Versorgung durchsucht, um ein neues Modell durch Synthese der gefundenen Ansätze aufzubauen [52]. Es wurden unterschiedliche aktuelle Ansätze zur terminologischen Fundierung von medizinischen Dokumentationssystemen, die sich der Fragestellung dieser Arbeit näherten bzw. deren Zielsetzungen gleich denen dieser Arbeit sind, eruiert, analysiert und bewertet. Die Literatur wurde auf dem Gebiet der terminologischen Forschung, der Versorgungs- und Studienforschung, und der Dokumentationssysteme im Gesundheitswesen untersucht. Bei der systematischen Suche wurde sich weder auf bestimmte Dokumentationssysteme, noch auf bestimmte Terminologien bzw. kontrollierte Vokabulare beschränkt. Die Literaturrecherche wurde zwischen Winter 2011 und Frühjahr 2012 über das Internet durchgeführt und fokussierte sich auf Veröffentlichungen aus medizinischen Datenbanken. Das Medical Literature Analysis and Retrieval System Online (MEDLINE, s. <http://www.ncbi.nlm.nih.gov/pubmed>) wurde hauptsächlich über PubMed durchsucht. ScienceDirect, eine der größten Online-Sammlungen von veröffentlichten wissenschaftlichen Forschungsarbeiten weltweit, die vom Elsevier Verlag betrieben wird, wurde ebenfalls durchsucht (s. <http://www.sciencedirect.com/>). Folgende wichtigsten Schlüsselwörter bzw. Suchphrasen wurden bei der Suche eingegeben: terminology-based documentation system, terminological support of documentation systems, terminological support AND documentation systems, terminologies in documentation systems. Mehr als 400 wissenschaftliche Arbeiten, die sich mehr oder weniger mit dem Thema befassten, wurden in den Bereichen Bioinformatik, Immunologie, Genetik, Klinische Pharmakologie, medizinische Informatik, etc. gefunden. Ein Teil der wissenschaftlichen Arbeiten führte eine Analyse von medizinischen Dokumentationssystemen durch und betonte in den Ergebnissen den Bedarf an Ontologie-basierten Mechanismen bzw. an stärkeren Terminologie-Systemen, um die Beschreibung/Aufzeichnung von Patientendaten bzw. von klinischen oder

administrativen Daten in der Versorgung und Forschung zu erleichtern und die Granularität und Genauigkeit dieser Daten zu verbessern. Ein anderes Teil stellte Mapping-Ergebnisse zwischen unterschiedlichen Klassifikations- und Terminologie-Systemen vor oder Ergebnisse von Usability-Bewertungen von Terminologie-basierten Frameworks. Die zeitaufwändige Suche wurde auf vier repräsentative Ansätze beschränkt, die den Zielsetzungen der vorliegenden Arbeit am besten entsprachen und die vor allem eine detaillierte Beschreibung des entwickelten Terminologie-basierten Dokumentationssystems lieferten. Um die Motivation der vorliegenden Arbeit jedoch zu rechtfertigen, wurden ebenfalls wissenschaftliche Arbeiten berücksichtigt, die die Notwendigkeit von standardisierten Dokumentationssystemen und die wichtige Rolle von medizinischen kontrollierten Vokabularen in diesem Standardisierungsprozess betonen.

In dem Top-Down-Ansatz wurde die Suche auf ein bestehendes Metamodell, nämlich auf das Metamodell des ISO/IEC 11179 Standards, beschränkt, aufgrund seiner Empfehlung vom höchsten Standardisierungsgremium für die Spezifikation von Metadaten und Verwaltung von Klassifikationssystemen (s. <http://metadata-standards.org/>).

Alle bestehenden Lösungen zur terminologischen Fundierung von Dokumentationssystemen werden in diesem Kapitel beschrieben. Um sie jedoch bewerten zu können, wurden Analyse- und Qualitätskriterien festgelegt. Diese betreffen hauptsächlich die funktionalen Anforderungen, sowie die Modellierung dieser Ansätze. Während Kapitel C die eruierten Ansätze vorstellt und die Qualitätskriterien aufführt und definiert, widmet sich Kapitel D der Bewertung dieser Ansätze anhand der festgelegten Analyse- und Qualitätskriterien.

## **C.2 TOP-DOWN-ANSATZ**

Für die vorliegende Arbeit wurde das Metamodell von ISO/IEC 11179 draft Version 3 “Information Technology - Metadata Registries (MDR)” Teil 3 „Registry Metamodel and basic attributes“ (im Folgenden ISO/IEC 11179 abgekürzt) evaluiert [50, 51, 53, 65].

“ISO/IEC 11179 describes the method of standardizing and registering of data elements to make data understandable and sharable. Its key concept is the data element, a unit of data for which the definition, identification, representation, and permissible values are specified by means of a set of attributes.” [31, s. auch <http://www.metadata-standards.org> für Einzelheiten über den ISO/IEC Standard]. Elemente des ISO/IEC 11179 Standards werden im Folgenden kursiv geschrieben.

Der ISO/IEC 11179 Standard beschreibt also die Methode zur Standardisierung und Registrierung von Datenelementen, um Daten sowohl syntaktisch als auch semantisch austauschen zu können. Somit wird Interoperabilität – sowohl semantische als auch syntaktische – zwischen heterogenen Systemen unterstützt. Der Standard definiert Regeln, sowie ein Metamodell für den Aufbau eines Metadata Registry bzw. Metadata Repository, welches ein Informationssystem zur Registrierung, Speicherung und Verwaltung von Metadaten darstellt. Metadaten repräsentieren dabei Daten, die andere Daten (in der realen Welt, unabhängig vom Medium (Papier oder elektronisch)) definieren oder beschreiben.

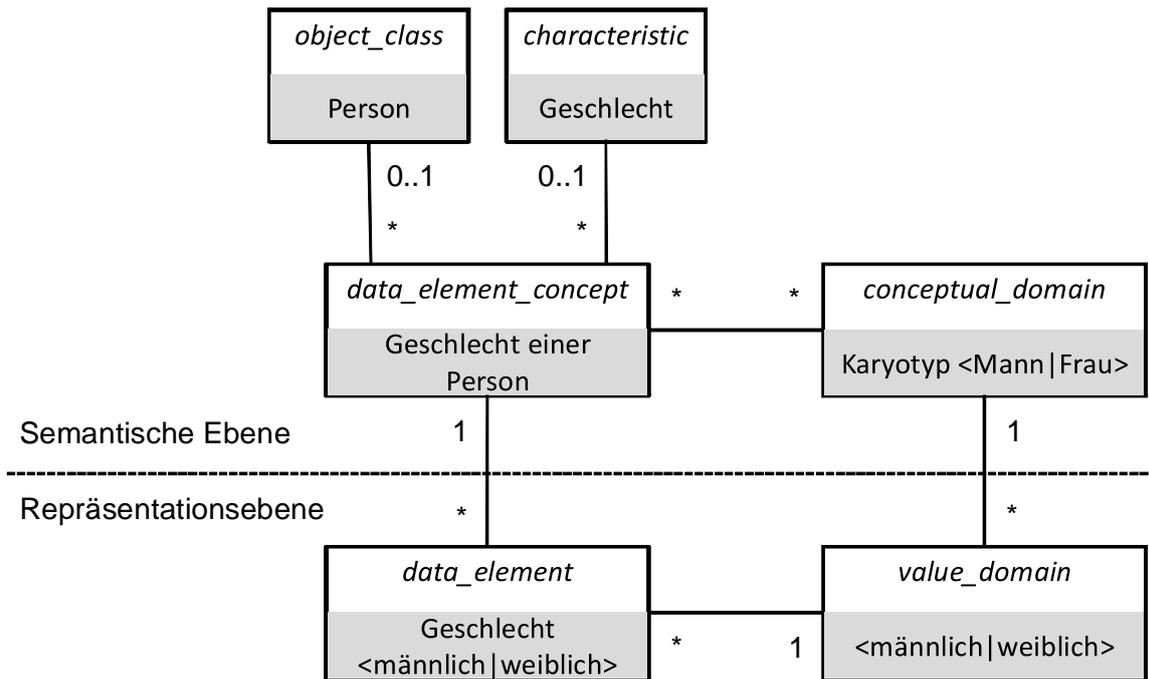
Noch spezifischer gesprochen sind Metadaten „Daten, die über die Art von anderen Daten, ihren Speicherort, was sie bedeuten, welche Organisationen für ihre Aufhebung und ihre Qualität verantwortlich sind, und wie man auf diese Daten zugreifen kann, berichten.“ [4, 33, 56]

Für die Registrierung, Definition, Beschreibung und Klassifikation eines Datenelements (*Data\_Element*), das zentrale Metadatum in ISO/IEC 11179, stellt der ISO/IEC 11179 Standard verschiedene Datenmodelle zur Verfügung. Das *Data Description Package* and das *Concepts Package* sind zwei Datenmodelle des ISO/IEC 11179 Metamodels, die für die vorliegende Arbeit Relevanz gefunden haben (s. Anhang K.1). Diese zwei Datenmodelle unterstützen nämlich jeweils die Beschreibung eines Datenelements mit seinen Wertebereichen, sowie die Modellierung und Einbindung externer Standards für die Klassierung und Standardisierung eines Datenelements.

Das *Data Description Package* stellt Metadaten für die semantische Beschreibung und die Darstellung (Repräsentation) von Datenelementen bereit. Das *Concepts Package* besteht aus der *Concept System Metamodel Region* und der *Classification Metamodel Region*. Ziel der *Concept System Metamodel Region* ist die Beschreibung von Konzepten aus externen Standards und ihren Beziehungen zueinander. Diese externen Standards können u.a. Taxonomien, Ontologie oder kontrollierte Vokabulare sein. Ziel der *Classification Metamodel Region* ist die Modellierung von Klassifikationsschemas durch die Benutzung von Konzepten aus der *Concept System Metamodel*.

Als Beispiel zur Veranschaulichung zeigt Abbildung 1 wie das *Data Description Package* des ISO/IEC 11179 Metamodels die semantische Ebene von der Darstellungsebene bei der Definition von Merkmalen unterscheidet. Eine semantische Ebene könnte zum Beispiel durch die Identifizierung von Männern und Frauen als zwei Arten von Personen mit unterschiedlichen Karyotypen (*Conceptual\_Domain*) gebildet werden. Das Konzept (*Data\_Element\_Concept*) „Geschlecht“ wird dabei mit dem Objekt (*Object\_Class*) „Person“ und der Eigenschaft (*Characteristic*) „Geschlecht“ gebildet. Die *Value\_Domain* definiert den Wertebereich bzw. die Werteliste für Objekte des Typs *Data\_Element\_Concept* und umfasst daher alle möglichen Werte (*Permissible\_Value*), die ein „Geschlecht“ annehmen kann, d.h. „männlich“ und „weiblich“. Letztendlich wird das eigentliche Dokumentationsmerkmal (*Data\_Element*) durch die Kombination des *Data\_Element\_Concept* „Geschlecht“ mit der *Value\_Domain* <männlich|weiblich> gebildet. Eine reale Instanz eines *Data\_Element* in diesem Beispiel wäre: „Ngouongo, Geschlecht: weiblich“.

Das ISO-Element *Conceptual\_Domain* hat zwei Spezialisierungen, *Enumerated\_Conceptual\_Domain* und *Described\_Conceptual\_Domain*, die jeweils aus einer endlichen Anzahl von eindeutigen *Value\_Meaning* (Konzeptsemantiken, Beispiel: {Mann, Frau}) und aus Beschreibungen, Regeln oder Referenzen (Beispiel: Alle Wesen auf der Erde) bestehen. Gleichermäßen hat das ISO-Element *Value\_Domain* zwei Spezialisierungen, *Enumerated\_Value\_Domain* und *Described\_Value\_Domain*, die jeweils aus einer endlichen Anzahl von eindeutigen *Permissible\_Value* (zulässige Werte, Beispiel: {männlich, weiblich}) und aus Beschreibungen oder Regeln (Beispiel: Ganzzahlen, Dezimalzahlen, etc.) bestehen (s. Anhang K.1).



**Abbildung 1: Grobe Darstellung des Data Description Package mit Beispielen in grau [31]**

Im *Concept System Metamodel* werden die Vokabulare durch das ISO-Element *Concept\_System* dargestellt und ihre Konzepte bzw. Klassen/Begriffe durch das ISO-Element *Concept*. Die (hierarchischen) Beziehungen zwischen Konzepten werden durch die ISO-Elemente *Link*, *Link\_End* und *Relation\_Role* dargestellt. Als Beispiel: die Krankheitsklasse (*Concept A*) "Cholera" mit dem Code A00.- aus dem *Concept\_System* "ICD-10-GM 2013" ist eine "infektiöse Darmkrankheit" (*Concept B*) mit dem Code A00-A09. Beide Krankheitsklassen werden als *Concept* registriert. Zwischen beiden Krankheitsklassen herrscht jeweils eine einfache hierarchische Beziehung – nach UML Spezialisierung und Generalisierung genannt –, die durch den *Link* "Hierarchie" und jeweils die *Relation\_Role* "is\_a" (Spezialisierung) und "inverse\_isa" (Generalisierung) abgebildet wird. (*Concept A is\_a Concept B* und *Concept B inverse\_isa Concept A*). Ein *Link\_End* ist nichts anderes als ein *Concept*, der in einer Beziehung zu einem oder mehreren anderen *Concepts* steht (s. Anhang K.1).

### C.3 BOTTOM-UP-ANSATZ

#### C.3.1 TERMTrial (*Terminology-based Remote-Data-Entry-System for clinical trials*)

TERMTrial ist ein Anwendungssystem, das die Entwicklung von Terminologie-fundierten Remote-Data-Entry-Systemen (RDE-Systemen) unterstützt [47]. Im Folgenden wird das Dokumentationsmodell von TERMTrial anhand seiner Anforderungen und Architektur näher vorgestellt. Modellelemente von TERMTrial werden kursiv geschrieben.

Die Behandlung von Patienten erfolgt immer öfter institutionsübergreifend. Folglich müssen dokumentierende Zentren zeitaufwändige Datenerhebungen und Auswertungen bewältigen, die durch unterschiedliche klinische Terminologien erschwert werden. Die Bereitstellung einer

institutionsübergreifenden klinischen Referenzterminologie würde Dokumentationsarbeiten erleichtern und eine standardisierte Dokumentation fördern, welche eine Voraussetzung für effiziente Kommunikation und Datenaustausch darstellt. TERMTrial wurde daher unter der Federführung der deutschen Gesellschaft für Pädiatrische Onkologie und Hämatologie (GPOH) entwickelt, um „ein Datenmodell zur Definition einer institutionsübergreifenden einheitlichen klinischen Terminologie“ bereitzustellen, sowie um ein rechnerbasiertes Terminologie-Managementsystem zur Verwaltung der klinischen Referenzterminologie zu verwalten [47]. Da heutige Dokumentationssysteme für klinische Studien vorzugsweise als RDE-Systeme entwickelt werden, kombiniert TERMTrial deren Vorteile (höhere Datenqualität, unmittelbare Dokumentation nach Patientenversorgung) mit einer übergreifenden terminologischen Steuerung. Somit wird eine einheitliche Nutzung eines festgelegten Terminologie-Systems in Studiendokumentationssystemen gewährleistet und dadurch werden Studiendokumentationssysteme verbessert. Nach Analyse von Dokumentationsaktivitäten und Entwicklungsprozessen von Dokumentationssystemen in zwei Zentren für klinische Studien der GPOH wurden folgende Aufgaben zur Entwicklung eines Terminologie-basierten Dokumentationssystems definiert:

- Ein Terminologie-Managementsystem für RDE definieren, in dem terminologische Konzepte/Begriffe, ihre Bezeichnungen, ihre Beziehungen untereinander, Merkmale und Einheiten beschrieben und verwaltet werden
- Eine allgemeine Referenzterminologie für alle multizentrischen klinischen Studien definieren und verwalten
- Eine relationale Studiendatenbank entwickeln, die mit der festgelegten Studienterminologie fundiert
- Terminologisch fundierte elektronische Case Report Forms (eCRFs) entwickeln

Merzweiler schlägt eine globale Architektur eines Anwendungssystems für die Entwicklung von terminologisch fundierten Dokumentationssystemen für kooperative Gruppen von multizentrischen Studien vor. Die Architektur von TERMTrial basiert auf seinen Anforderungen und seine Entwicklung beruht auf einem Datenbank-gerichteten Ansatz. Bei einem Datenbank-gerichteten Ansatz gibt es kein Mapping auf der CRF-Ebene, d.h. keine direkte Zuordnung zwischen CRF-Elementen und Datenbank-Tabellen, sondern auf der Ebene der terminologischen Kontexten (Ebene der Semantik). Das Schema der Datenbank hängt somit nicht vom Inhalt und der Struktur der Fragebögen ab. Das Datenmodell von TERMTrial besteht aus einer Komponente für die Definition und Verwaltung von Terminologie-Systemen für Kooperativgruppen von klinischen Studien und aus zwei Komponenten für die automatische Generierung von terminologisch fundierten Studiendatenbanken und den interaktiven Entwurf von (eCRFs) [47].

Diese drei zentralen Komponente von TERMTrial werden im Folgenden namentlich und im Einzelnen beschrieben (s. Abbildung 2):

1. Das Terminologie-Management-System (*Terminology Management System, TMS*) dient der Bereitstellung und Verwaltung von allgemeinen (*Common Terminology Module, CTM*) und studien- bzw. institutionsspezifischen (*Trial Specific Terminology Module, TSTM*) Terminologie(-Systemen). *TSTM* ist nichts anderes als eine Spezialisierung bzw. Instanziierung des *CTM*.
2. Die Datenbank-Generierung-Komponente (*Database Generating Component, DBGC*) dient der automatischen Generierung von terminologisch fundierten Studiendatenbanken aus dem Studiendatenmodell (*Trial Data Model*), indem sie Informationen aus dem Terminologie-System transformiert und ein SQL-DDL-Skript generiert, das von einem SQL-Server ausgeführt werden kann.
3. Die Formular-bauende-Komponente (*Form Building Component, FBC*) dient der Erzeugung von terminologisch fundierten elektronischen Eingabefeldern.

Es lässt sich folgendes kleines Anwendungsszenario beschreiben (vgl. Abbildung 2): Nachdem ein Koordinierungszentrum für klinische Studien sein Terminologie-System und sein Studiendatenmodell im *TMS* spezifiziert und definiert hat, kann darauf basierend eine neue Studiendatenbank mit der *DBGC* automatisch generiert werden. Anschließend können gewünschte Datenelemente mit der *FBC* interaktiv in logischen Formularen gruppiert und darauf aufbauend die eCRFs erzeugt werden. Für die logische Darstellung von eCRFs stellt das Studiendatenmodell Konzepte (*Object types/Context*) und ihre Charakteristiken (*Items/Trial Specific Items*) in einer Baumstruktur bereit. Mit der neuen Studiendatenbank wird ebenfalls ein Data Dictionary (DD) generiert, das nötige Informationen für die automatische Generierung von SQL-Funktionen für eCRFs beinhaltet. Ein Data Dictionary ist in diesem Kontext von relationalen Datenbanken eine Datenbank, die Metadaten enthält und von einer Abfragesprache (hier SQL) nur gelesen wird.

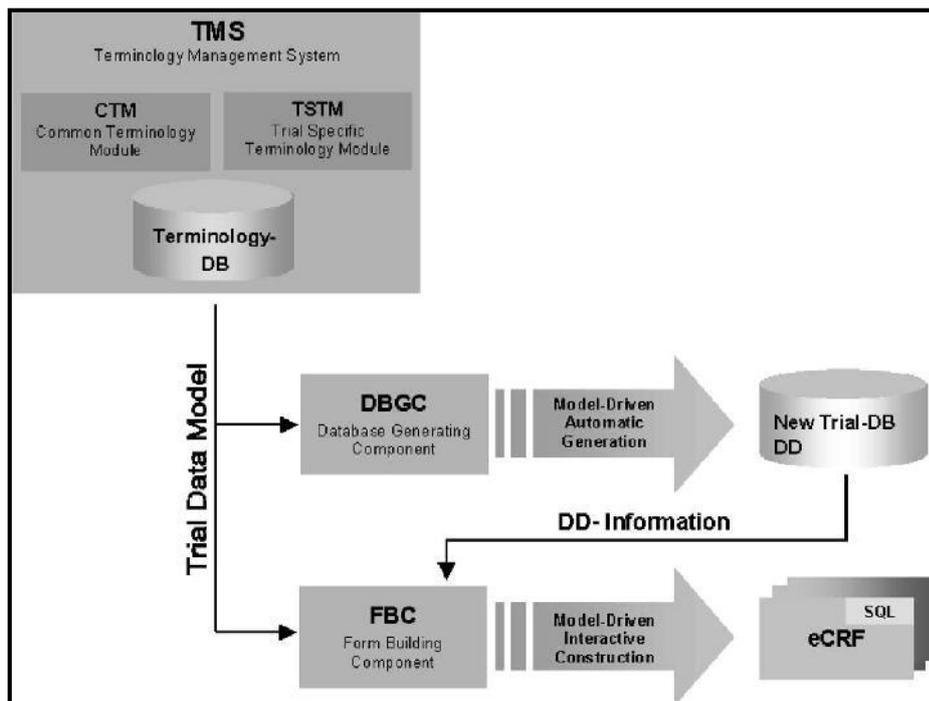


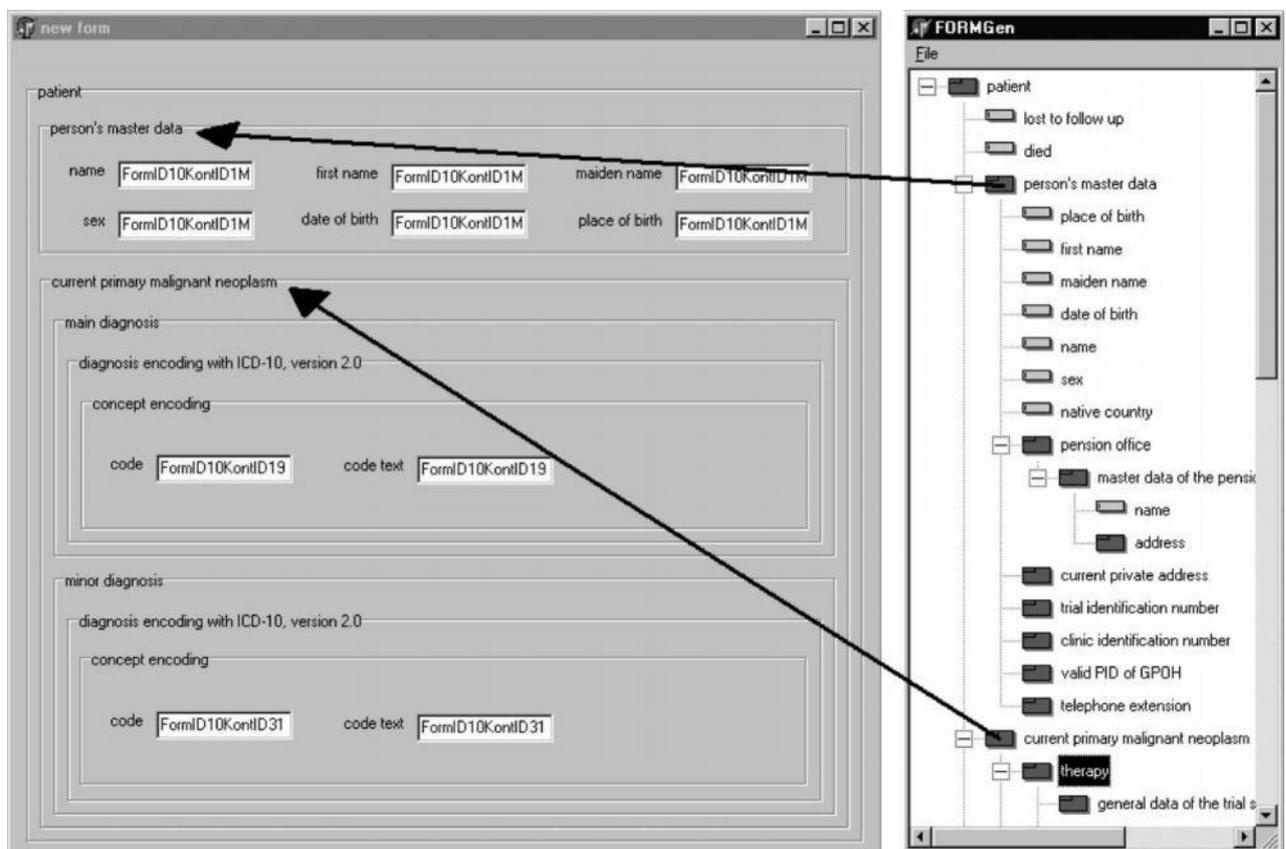
Abbildung 2: Systemarchitektur von TERMTrial [47]

Folgende Kernmodellelemente definieren ein allgemeines Terminologie-System (CTM) im Sinne Merzweilers (s. Anhang K.1):

- *Concepts*: *Concepts* sind medizinische Begriffe (aus Terminologien). Für jeden *Concept* können synonyme Begriffe, textliche und formale Definitionen, und dessen Beziehungen zu anderen *Concepts* verwaltet werden. *Concepts* definieren Merkmale eines Dokumentationssystems.
- *Object type*: ein Objekttyp ist ein *Concept*, das den Typ eines Dokumentationsmerkmals definiert. Beispiel: „Krankheit“. Ein *Concept* kann wiederum als *Object type* benutzt werden.
- *Terms*: ein Term definiert ein Synonym oder eine andere Bezeichnung eines *Concepts*. Ein *Term* kann ebenfalls für die Bezeichnung von Datenbankelementen benutzt werden.
- *Relations*: es gibt zwei Arten von hierarchischen Beziehungen zwischen *Concepts*: generische Beziehungen (is\_a) und Part-of-Beziehungen (is\_part-of).
- *Items*: *Items* definieren Charakteristiken von *Object types*. Beispiel: das Dokumentationsmerkmal „Relevanz einer Krankheit“ setzt sich aus dem *Object type* „Krankheit“ und dem *Item* „Relevanz“.
- *Option*: Besitzt ein *Item* den Datentyp *Option*, so soll eine Wertemenge (*set of options*) als Auswahlliste für dieses *Item* definiert werden. Beispiel: Für den *Object type* „Krankheit“ soll die Eigenschaft (*Item*) „Relevanz“ dokumentiert werden. Folgende mögliche Werte werden zur Auswahl angeboten (*set of options*): {primär, sekundär, andere Relevanz}. Jede *Option* kann ein Mapping zu mehreren *Concepts* im CTM besitzen.

Im *TSTM* wird das Terminologie-System jeder klinischen Studie definiert und mit seiner zugrunde liegenden Terminologie im *CTM* verlinkt. Folgende Elemente können im *TSTM* spezifiziert und administriert werden (s. Anhang K.1):

- *Contexts*: als Spezialisierungen von *Object types* definieren *Contexts* Objekttypen für studienspezifische Merkmalsarten (*Trial Specific Items*). *Contexts* können untergeordnete bzw. übergeordnete *Contexts* haben. Die Kardinalität zwischen *Contexts* ist daher zu spezifizieren.
- *Trial Specific Items*: *Trial Specific Items* sind als Instanzen von allgemeinen *Items* studienspezifische Dokumentationseinheiten (Merkmalsarten). Für *Items* vom Datentyp *Option* ist eine Wertemenge (*set of options*) für die *Trial Specific Items* zu definieren (siehe *Items*).
- *Clinical trials*: Das Element *Clinical trial* im *TSTM* definiert eine klinische Studie und setzt sich aus einer Menge von *Contexts* – werden von der klinischen Studie benutzt – und einer Menge von *Trial Specific Items* – müssen in den *Contexts* erfasst werden – zusammen (s. Abbildung 3).



**Abbildung 3: Form Building Component [47]. Die Zusammenstellung von eCRF-Elementen erfolgt per Drag and Drop von rechts (Baumstruktur) nach links (eCRF) (s. Pfeile).**

### C.3.2 InCoMe (Integriertes System zur Computerunterstützten Entwicklung medizinischer Dokumentationssysteme)

InCoMe ist ein integriertes System zur computerunterstützten Entwicklung von medizinischen Dokumentationssystemen, welches eine Grammatik zur Modellierung von medizinischen

Dokumentationssystemen unter besonderer Berücksichtigung von Integritätsbedingungen benutzt, um die Integrität und Vollständigkeit der gesammelten Daten zu erhöhen [20, 21].

Im Folgenden wird das Dokumentationsmodell InCoMe anhand seiner Anforderungen und Architektur näher vorgestellt. Modellelemente von InCoMe werden zwischen Anführungszeichen geschrieben.

InCoMe wurde 2005 in der Abteilung Medizinische Informatik der Universität Duisburg-Essen von Goertzen entwickelt. Die Hauptmotivation von Goertzen bei der Entwicklung von InCoMe war, einen Beitrag zur Verbesserung der Qualität der Patientenversorgung durch computerunterstützte Dokumentation zu leisten. Es geht aus seiner Dissertation hervor, dass die Qualität medizinischer Daten einen wesentlichen Aspekt bei deren Erhebung und Verwertung darstellt, was unsere Erkenntnis bestätigt. Ein rechnergestütztes medizinisches Dokumentationssystem sollte folglich vor allem Anwender bei einer vollständigen und plausiblen Erfassung aller relevanten Daten unterstützen. InCoMe sollte ebenfalls die Wiederverwendung von medizinischen Merkmalen unterstützen, sowie den Arbeitsaufwand bei der Definition von Integritätsbedingungen und Dokumentationssystemen reduzieren [20]. InCoMe wurde entworfen, entwickelt und getestet unter Verwendung des Merkmalskataloges für Erwachsene aus dem Kompetenznetz HIV/AIDS.

Goertzen stellt Konzepte in seiner Dissertation vor, „die eine durchgängige und übergreifende Sicherung der Qualität in medizinischen Dokumentationssystemen auf Modell-, Prozess- und Software- Engineering-Ebene unterstützen. In diesem Zusammenhang steht nicht die Datenqualität allein im Fokus der Betrachtung, sondern vielmehr auch die Qualität des Systems als Ganzes, die durch geeignete Vorgehensmodelle und Techniken bei Entwicklung und Implementierung sicherzustellen ist.“ [20, 21]

Entscheidend bei InCoMe ist das zugrunde liegende plattformunabhängige Dokumentationsmodell, das sich gegen ein definiertes XML-Schema validieren lässt und somit allen im Schema definierten Regeln folgt. Dieses Dokumentationsmodell beschreibt das zu generierende Clinical Data Management System (CDMS, klinisches Dokumentationssystem) mit folgenden sechs hierarchisch gegliederten Hauptelementen: „Applikation“, „Datenbank“, „Kontext“, „Merkmalsart“, „Stammdaten“ und „Integritätsbedingung“; wobei das Modellelement „Applikation“ sich aus allen anderen Modellelementen zusammensetzt und somit auf der obersten Ebene der Elementen-Hierarchie steht (s. Abbildung 4 und Anhang K.1).

Merkmalsarten sind Eigenschaften von Dokumentationseinheiten. Man unterscheidet zwei Typen von Merkmalsarten: freie und gebundene Merkmalsarten. Eine „freie Merkmalsart“ definiert eine allgemeine übergreifende Eigenschaft (z.B. Name, Geburtsdatum, Krankheit), die jederzeit und mehrfach im Dokumentationsmodell verwendet werden kann. Sobald sie einem „Kontext“, also einem Merkmalsträger (z.B. Patient), direkt zugeordnet wird, wird sie zu einer „gebundenen Merkmalsart“ (z.B. Name eines Patienten).

„Stammdaten“ definieren eine wiederverwendbare finite Menge von Merkmalsausprägungen, die die möglichen Werte einer „Merkmalsart“ einschränken (z.B. Krankheit: {Cholera, Influenza, Malaria, Fieber, Heuschnupfen, etc.}).

Ein „Kontext“ bindet Merkmalsarten, die in einer semantischen Beziehung zueinander stehen (z.B. alle Merkmalsarten einer spezifischen Dokumentationseinheit), und eine „Datenbank“ bündelt alle semantisch zusammengehörende Kontexte.

„Merkmalsarten“, „Kontexte“ und „Datenbanken“ haben „Integritätsbedingungen“, die die Integrität (Korrektheit und Vollständigkeit) der im medizinischen Dokumentationssystem erfassten Daten sicherstellen. Der Schwerpunkt bei dem Dokumentationsmodell von Goertzen liegt in der Formulierung von Integritätsbedingungen der verwalteten Daten, da sie die Datenqualität sicherstellen. Im Dokumentationsmodell von Goertzen wird auf die Formulierung von zehn unterschiedlichen Ausprägungen von Integritätsbedingungen eingegangen.

- Datentyp-Integritätsbedingung: hier wird der Datentyp einer Merkmalsart festgelegt (z.B. die Merkmalsart <Geburtsdatum> soll vom Datentyp <Date> sein.)
- Datenlänge-Integritätsbedingung: hier wird die Länge einer Merkmalsart festgelegt (z.B. die Merkmalsart <Nachname> soll vom Datentyp <varchar> sein und die Länge 45 nicht überschreiten.)
- Wertebereich-Integritätsbedingung: hierdurch wird der Bereich von Werte beschränkt, die eine Merkmalsart annehmen kann. Z.B. die Merkmalsart <diastolischer Blutdruck> darf nur Werte zwischen 70 und 90 annehmen. Der Wertebereich wäre also [70-90].
- Schlüssel-Integritätsbedingung: dieser Integritätsbedingung ähnelt die Primary Key Integritätsbedingung in einer relationalen Datenbank. Gebundene Merkmalsarten mit dieser Integritätsbedingung identifizieren nämlich eindeutig den zugehörigen Kontext. Z.B. die Merkmalsart Patientennummer identifiziert einen Patienten eindeutig.
- Anzahl-Integritätsbedingung: dieser Integritätsbedingung ähnelt die Kardinalität Integritätsbedingung in einer relationalen Datenbank. Es wird hier nämlich festgelegt wie viele Merkmalsausprägungen einer gebundenen Merkmalsart erfasst werden können. Z.B. die gebundene Merkmalsart <Geburtsdatum> eines Patienten muss bei der Erhebung einmal erfasst werden (Pflichtmerkmalsart; Anzahl(min, max)=(1,1). Die gebundene Merkmalsart <Familienstand> eines Patienten kann, muss aber nicht erhoben werden (optionale Merkmalsart; Anzahl (min, max)=(0, 1)). Die letzte mögliche Variante ist (0,0). In diesem Fall darf die Merkmalsart keine Ausprägung haben.
- Referenz-Integritätsbedingung: dieser Integritätsbedingung ähnelt die Foreign Key Integritätsbedingung in einer relationalen Datenbank. Eine gebundene Merkmalsart mit dieser Integritätsbedingung referenziert auf eine andere gebundene Merkmalsart.

- **Suche-Integritätsbedingung:** eine gebundene Merkmalsart mit dieser Integritätsbedingung wird als Suchfeld in einer Suchoperation im Dokumentationssystem definiert.
- **Ergebnis-Integritätsbedingung:** eine gebundene Merkmalsart mit dieser Integritätsbedingung darf nach einer Suchoperation im Dokumentationssystem in den Suchergebnissen angezeigt werden.
- **Hierarchie-Integritätsbedingung:** hierdurch wird festgelegt, wie zwischen den modellierten Kontexten zu navigieren ist. Somit lässt sich der Aufbau des Dokumentationssystems festlegen.
- **Semantische Integritätsbedingung:** hier werden semantische Regeln über die zu dokumentierenden medizinischen Daten definiert. Z.B. die Merkmalsart <Schwangerschaft> darf bei einem männlichen Patienten nicht den Wert ‚Ja‘ haben.

Verletzungen dieser Integritätsbedingungen werden im Dokumentationssystem entsprechend durch Fehler- bzw. Warnungsmeldungen behandelt.

Die Architektur von InCoMe besteht aus folgenden drei weiteren wichtigen Strukturen zur Generierung des CDMS (s. Abbildung 5):

**Das Entwurfsmodul (CDMS-Designer)** stellt eine grafische Benutzeroberfläche zur Erstellung und Erzeugung eines konkreten Dokumentationsmodells dar. Die Dokumentationsmodellierung mit dem Entwurfsmodul erfolgt nicht durch den Endbenutzer (Arzt, Pflegepersonal, Studienpersonal, etc.), sondern durch den Systemdesigner bzw. Datenmanager. Der Endbenutzer benutzt lediglich das generierte Dokumentationssystem.

**Das Generierungsmodul (CDMS-Creator)** stellt Mittel zur Verfügung, die die Transformation des entworfenen konkreten Dokumentationsmodells in andere Darstellungen ermöglichen.

Auf Basis eines Systems aus Applikationsservern, Servlets, Java Server Pages, Struts, sowie relationalen Datenbanken wurde ein integratives Framework bereitgestellt. Über dieses **Applikationsframework** erfolgt die Überführung jedes Entwurfs eines konkreten Dokumentationsmodells direkt, ohne Quellcode-Kompilierung, in eine lauffähige Web-Anwendung nach vorheriger Datenextraktion in einem SQL-Statement, sowie die Erzeugung wichtiger Anwendung-Konfigurationsdateien. Dank der Datenbeschreibung wird die Datenbank des Dokumentationssystems generiert, und dank der Anwendungsbeschreibung wird die Logik des Dokumentationssystems aufgebaut.

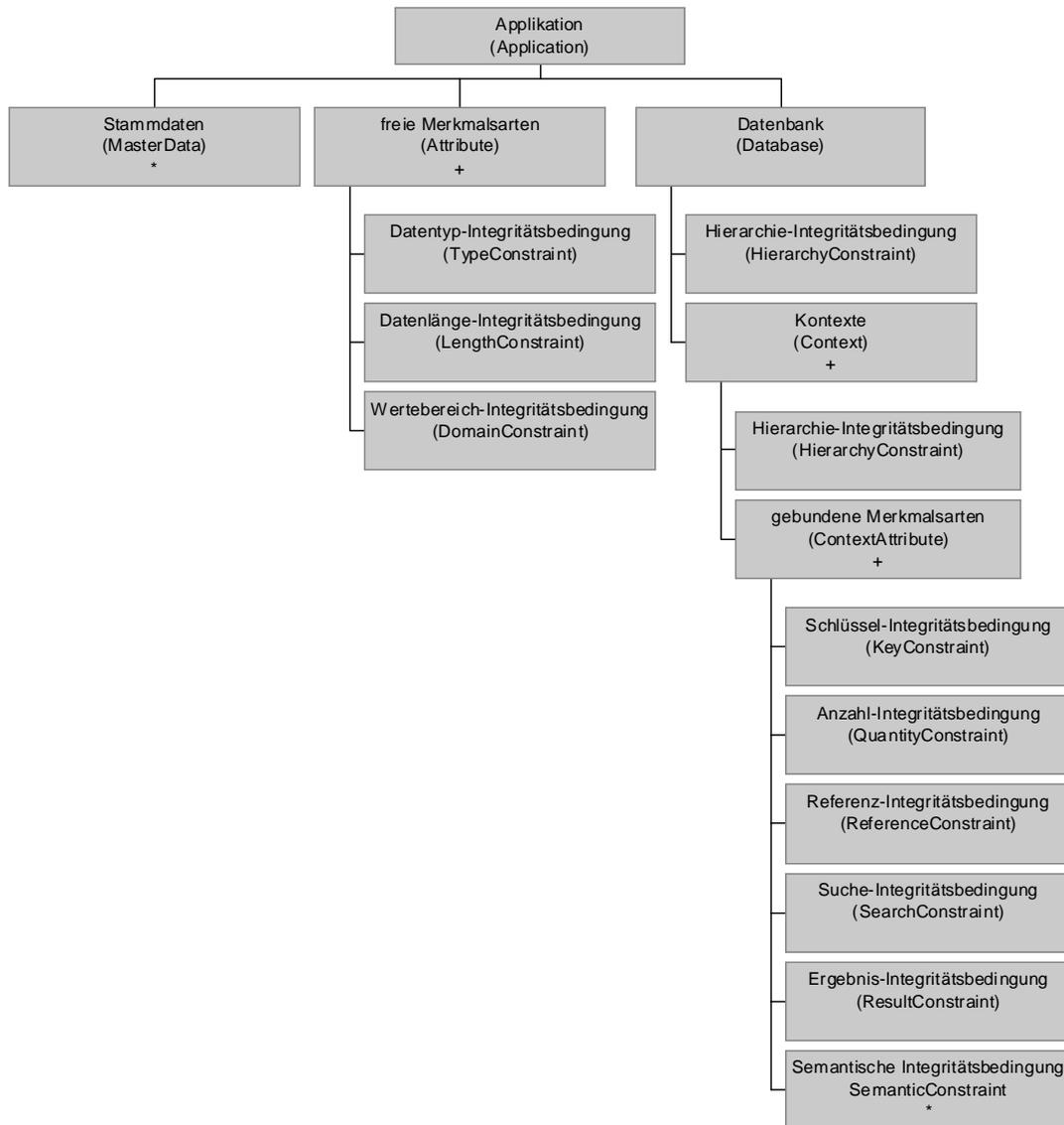


Abbildung 4: Dokumentationsmodell InCoMe [20]

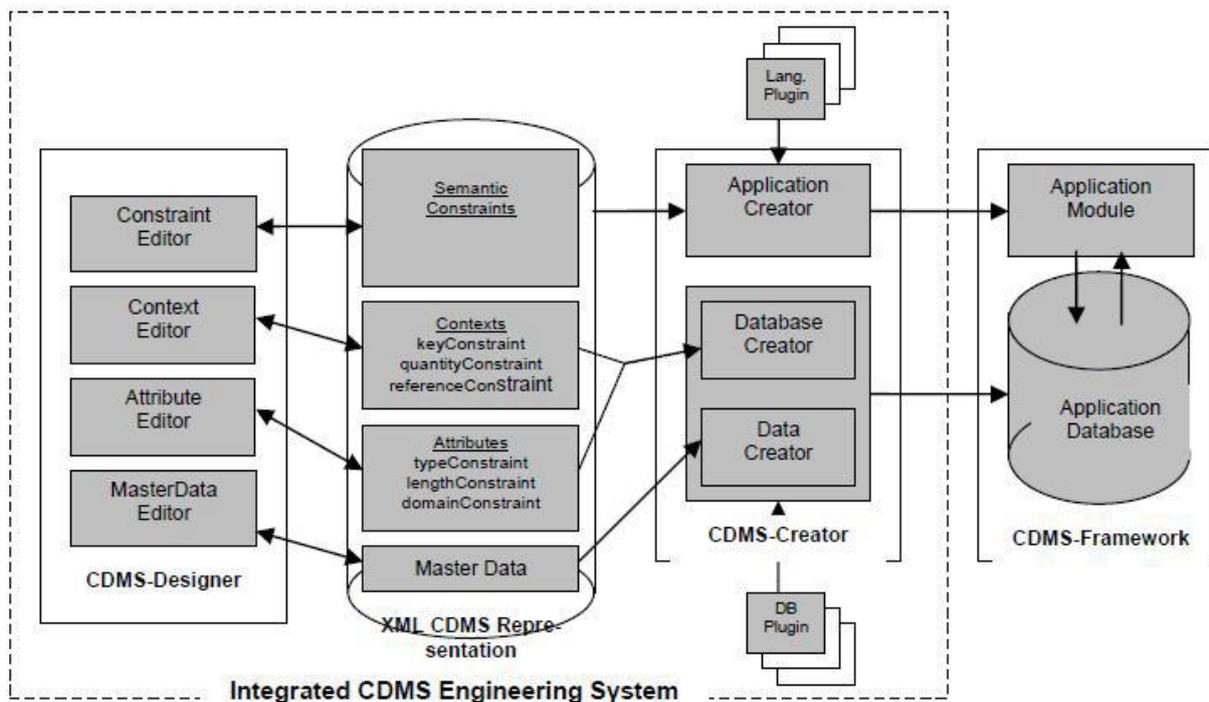


Abbildung 5: Architektur InCoMe [20]

### C.3.3 ORCA (Open Record for CARE)

Das generische, computerunterstützte Patientendokumentationssystem (CPR, Computer-based Patient Record) ORCA (Open Record for CARE) wurde 1999 in der Abteilung „Medizinische Informatik“ der Erasmus Universität Rotterdam in den Niederlanden entwickelt. Im Folgenden wird das Datenmodell des computerunterstützten Patientendokumentationssystems namens ORCA [18, 48] anhand seiner Anforderungen und Architektur näher vorgestellt.

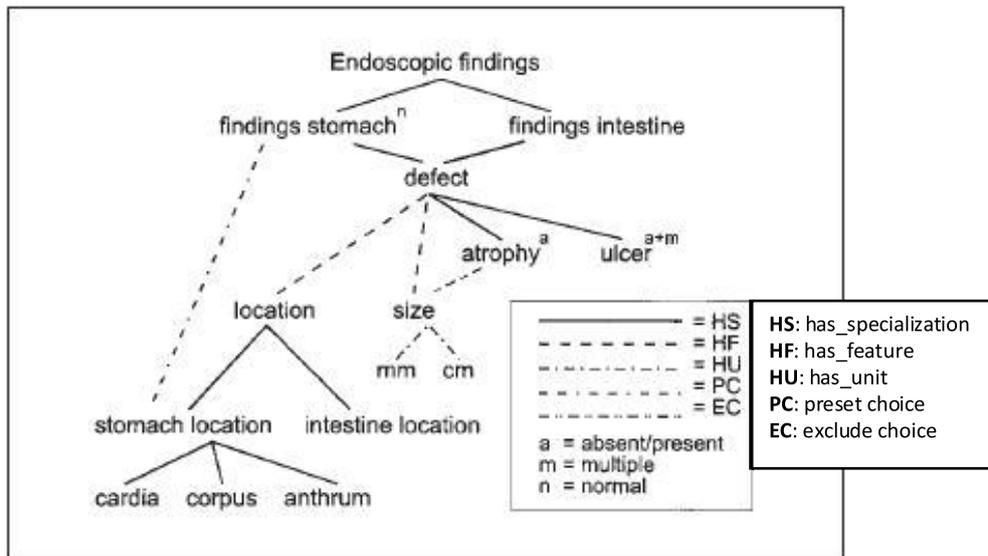
Wie Ginneken erkannt hat: Die wichtigsten Herausforderungen der heutigen Entwicklungen im medizinischen Bereich sind Datenintegration und Unterstützung strukturierter Dateneingabe (SDE, Structured Data Entry) in Versorgungszentren [18]. SDE zielt auf die Bereitstellung relevanter Informationen eines spezifischen medizinischen Kontexts ab. Ginneken bemängelt die Tatsache, dass trotz der Einführung von Computern im Gesundheitswesen Patientendaten teilweise immer noch papierbasiert zwischen Versorgungseinrichtungen übermittelt werden. Somit wird die Relevanz der semantischen und syntaktischen Interoperabilität in Datenübermittlungen unterstrichen. Außerdem hebt Ginneken hervor, dass die Sammlung von strukturierten medizinischen Daten viele Vorteile für Forschung, Entscheidungsunterstützung und Qualitätsprüfung bietet [18]. ORCA wurde zur Integration mit bestehenden Systemen und zur eindeutigen und konsistenten Beschreibung/Darstellung von Patientendaten und klinischen Befunden konzipiert. Zudem ermöglicht ORCA die Erfassung von strukturierten Patientendaten, die sich für Forschungszwecke, Entscheidungsunterstützung und institutionsübergreifende Versorgung eignen [19].

Folgende wesentlichen Anforderungen wurden beim Design und bei der Entwicklung von ORCA berücksichtigt: Eindeutige Darstellung von Daten, Flexibilität in Umfang, Inhalt und jeweiligem medizinischem Bereich, Anpassungsfähigkeit an unterschiedlichen Workflows, hohe Wartbarkeit.

ORCA verfolgt in seinem Datenmodell (s. Anhang K.1) einen wissensbasierten Ansatz (Knowledge-driven approach), um eine flexible, strukturierte Dateneingabe zu unterstützen. Hierfür wird eine Wissensbasis, die so genannte Knowledge Base (KB) oder auch Domain Model genannt, erzeugt. Die Knowledge Base ist für die SDE essenziell und enthält ein kontrolliertes Wörterbuch bzw. Lexikon von medizinischen Wissensbeschreibungen, die Begriffe definieren, welche im Patientendokumentationssystem verwendet werden können und die Regelungen festsetzt, wie diese Begriffe zu medizinisch sinnvollen Datensätzen miteinander kombiniert werden können. Die Knowledge Base besteht also aus einem Thesaurus von medizinischen Konzepten (*concepts*) und einem gerichteten Graphen, der diese Konzepte in Beziehung setzt bzw. definiert, wie die Konzepte zu verbinden sind (durch ihre Anordnung im Graph) und somit die strukturierte Dateneingabe unterstützt. Die sinnvolle Anordnung von Konzepten im Graph bildet den medizinischen Kontext. Die endständigen Elemente in der ORCA-Hierarchie sind die Deskriptoren (*descriptors*). Deskriptoren sind Eigenschaften bzw. Kinderelemente von Konzepten im Graph; sie beschreiben diese also in ihren Einzelheiten und sind ebenfalls in der Knowledge Base enthalten. Man beachte, dass ein *concept* auch als *descriptor* in mehr als einem bestimmten medizinischen Kontext verwendet werden kann.

Beziehungen zwischen Konzepten, wie sie in herkömmlichen konzeptuellen Graphen zu finden sind, werden in der Knowledge Base mit sechs Beziehungstypen abgebildet (s. Abbildung 6): *has\_specialization* (definiert Klasse/Subklasse-Hierarchien), *has\_feature* (spezifiziert ein Merkmal eines Konzepts), *has\_unit* (für quantitative Beschreibungen bzw. Angabe von Maßeinheiten), *refers\_to* (beschreibt die Referenz eines selben Konzepts in verschiedenen Kontexten), *preset choice* (für die automatische Auswahl eines Konzepts; *preset choice* Konzepte müssen nicht explizit dargestellt werden), *exclude choice* (schließt Kinder eines Konzepts aus, wenn dies ausgewählt wird). Der Datensatz "blood pressure – <has systolic pressure> – mmHg" wird beispielweise folgendermaßen in ORCA Knowledge Base modelliert: blood pressure – *has\_feature* – systolic blood pressure – *has\_unit* – mmHg. Neben diesen Beziehungen können Konzepte „Properties“ haben, die diese Konzepte charakterisieren. Properties können folgende Werte annehmen: „absence/presence“ (a), „single/multiple instantiations within the same context“ (m), und „normal“ (n) für Konzepte, deren Eigenschaften (*features*) vom Benutzer als normal betrachtet werden.

Nach Festlegung des Thesaurus und Entwurf der darauf basierten Knowledge Base wird diese im ORCA-Dokumentationssystem durch Import zugrunde gelegt. Erst dann fängt die strukturierte Patientendateneingabe an. Das ORCA-Dokumentationsmodell ermöglicht die Modellierung von verschiedenen medizinischen Bereichen ohne Einschränkungen auf das Beschreibungsniveau.



**Abbildung 6: Beispiel des Inhalts der Knowledge Base [18]. Beziehungen zwischen Konzepten sowie Eigenschaften wurden dargestellt. Semantiken bestehen aus Konzepten selbst und ihrer Anordnung.**

### C.3.4 ICNP-based ENRS (ICNP-based Electronic Nursing Record System)

Das ICNP-based Electronic Nursing Record System ist ein elektronisches Pflegedokumentationssystem, dessen Konzipierung die Pflegeabteilung des seoulischen nationalen Universitätsklinikums Bundang im Jahr 2002 vornahm.

Im Folgenden wird das Dokumentationsmodell des ICNP-basierten Electronic Nursing Record System (ENRS) anhand seiner Anforderungen und Architektur näher vorgestellt [55].

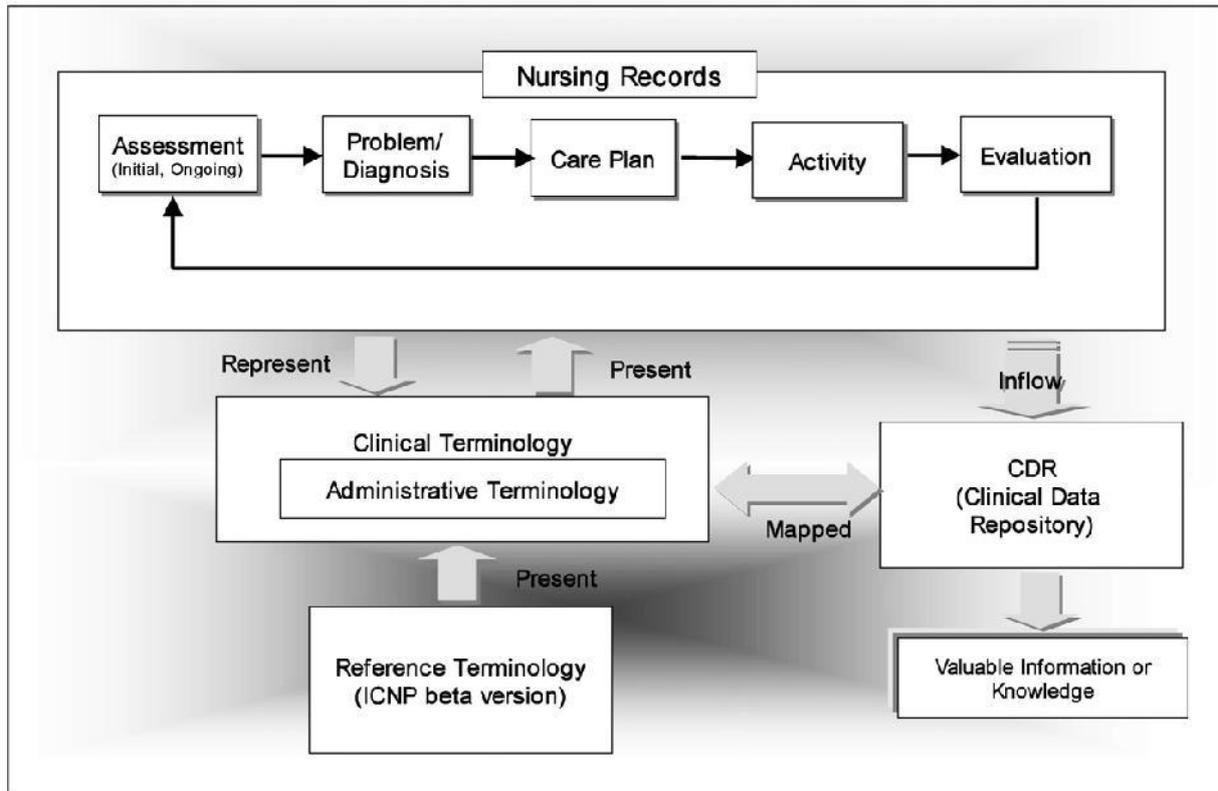
Mit stets wachsenden Anforderungen an die Pflege bedarf die Pflegedokumentation eines neuen Dokumentationsmodells zur Verwaltung von Pflegeinformationen. Das ENRS-Projekt wurde von der Pflegeabteilung des seoulischen nationalen Universitätsklinikums Bundang in Korea initiiert. Nach Park stellt ein Computer-basiertes Pflegedokumentationssystem die Grundlage einer neuen Art der Verwaltung von Pflegeinformationen dar und ermöglicht die Erfassung klinischer Informationen und deren Darstellung mit kontrollierten Terminologien, die weithin als notwendig in der medizinischen Dokumentation erachtet werden [55]. Park erkennt somit die verschiedenen potenziellen Vorteile kontrollierter Terminologien, u.a. die größere Ausdruckskraft und die umfangreichere Wiederverwendung von Daten aus heterogenen Datenquellen. Zudem erleichtern standardisierte Terminologien die Beschreibungen, Vergleiche und die Kommunikation von Aktivitäten der Pflegeversorgung zwischen Einrichtungen und Ländern. Die elektronische Pflegedokumentation sollte außerdem, ähnlich der elektronischen Patientenakte, eine ausnutzbare Quelle für die Wiederverwendung von Patientenprimärinformationen darstellen und nicht nur als einzige Data Repository für die Dokumentation von durchgeführten Pflegeaktivitäten dienen.

Das primäre Ziel des ENRS ist, die Nutzbarkeit von Pflegedokumentationssystemen in der Verwaltung von Pflegeinformationen zu maximieren, und somit dem Pflegepersonal zu ermöglichen, Versorgungs- bzw. Pflegeabläufe eines Patienten zu verwalten, indem Pflegeaktivitäten nachbetreut und alle Pflegeprozesse dokumentiert werden.

Für die Modellierung eines terminologisch fundierten elektronischen Pflegedokumentationssystems wurden die objekt-orientierte Analyse (OOA) und das objekt-orientierte Design (OOD) eingesetzt. Als modellgetriebener Ansatz wurde auf das Rational Unified Process (RUP) zurückgegriffen, um den Modellierungsprozess zu steuern. Die Unified Modeling Language (UML) wurde ebenfalls eingesetzt, um die Systemkomponenten und deren Verhalten entsprechend zu spezifizieren. Man spricht hier von Use-Case-Spezifikation. Zudem sollte das System eine Standard-Pflegeterminologie benutzen, um die Wiederverwendung von Pflegedaten zu unterstützen. Somit sollten ebenfalls u.a. die Datenqualität, die Entscheidungsunterstützung und der Vergleich von Pflegediensten verbessert werden. Die Wahl wurde auf die vom International Council of Nurses (ICN) entwickelte International Classification for Nursing Practice (ICNP) in seiner Beta-Version wegen seiner Ausdruckfähigkeit und seiner Flexibilität getroffen. Um Dokumentationsmerkmale, deren Arten und Ausprägungen zu identifizieren, wurden 69 Pflegeformulare aus 62 Pflegeabteilungen analysiert und auf Gemeinsamkeiten und Unterschiede geprüft.

Für die Entwicklung des terminologisch fundierten Pflegedokumentationssystems des seoulischen Universitätsklinikums wurde ein Pflegeinformationsmodell (s. Abbildung 7) erstellt. Das Modell besteht aus zwei Hauptdomänen zwischen denen Pflegedaten fließen. Der obere Teil des Modells stellt den Inhalt der Pflegedokumentation gemäß dem Pflegeprozess dar, und der untere Teil zeigt die Komponente und die Rollen eines Terminologie-Servers und ein klinisches Data Repository, welches klinische Informationen sichert. Die Anwendungsfälle des ENRS wurden folglich funktional in zwei Pakete geteilt: die Pflegedokumentation (Nursing Records) und das Pflegeterminologie-Management System (Nursing Term Management) (s. Anhang K.1).

Der Terminologie-Server verwaltet drei Arten von kontrollierten Terminologien: klinische, administrative und Referenzterminologien. Die ICNP wurde als Referenzterminologie in der Form von sogenannten „controlled nursing statements“ (Deutsch: kontrollierte Pflegedatensätze) angewendet. Diese Pflegedatensätze bestehen aus drei Datenkategorien, Pflegediagnosen, Pflegeaktivitäten und Pflegeberichte, und stellen semantische präkoordinierte Datensätze (precoordinated statements) aus ICNP-Konzepten dar. Aus der ICNP-Terminologie entsteht somit die klinische Terminologie des ENRS. Die administrative Terminologie ist eine Teilmenge der klinischen Terminologie und wird für die Analyse, Klassifikation und Zusammenfassung klinischer Daten im klinischen Data Repository benutzt.



**Abbildung 7: Pflegeinformationsmodell für das elektronische Pflegedokumentationssystem (ENRS) [55]**

#### C.4 ANALYSE- UND QUALITÄTSKRITERIEN

Aus den vier eruierten Lösungen im Bottom-Up-Ansatz ließen sich Ähnlichkeiten feststellen, was die Anforderungen eines terminologisch fundierten Dokumentationssystems angeht. Bei den meisten Lösungen war auch das Vorgehen ähnlich. Hinsichtlich der Entwicklung eines Terminologie-basierten Dokumentationssystems wurde aus der Literatur sichtbar, dass die Definition eines Terminologie-Systems zur Erleichterung der Definition und Beschreibung von medizinischen Dokumentationsmerkmalen ebenso wichtig ist wie die terminologische Steuerung eines Dokumentationssystems zur Standardisierung und Harmonisierung dieser Merkmale [26, 27, 37, 58, 59]. Die Nutzung eines Terminologie-Servers bzw. der Ansatz einer standardisierten Terminologie bzw. Klassifikation im entwickelten System war von entscheidender Bedeutung, um dem Dokumentationssystem ein Terminologie-Support anbieten zu können, sowie die standardisierte Darstellung und den Austausch von Dokumentationsmerkmalen/klinischen Daten zu ermöglichen. In allen Lösungen wurden ein IST-Zustand, sowie die anfallenden Dokumentationsaktivitäten in dem jeweiligen Klinikum/Institut analysiert, indem alle verwendeten CRFs und sonstige Merkmalskataloge im aktuellen verwendeten Dokumentationssystem analysiert wurden, um alle möglichen dokumentierten Informationstypen und verwendeten Datentypen zu identifizieren. Die Ergebnisse der Analyse der Merkmalswelt (Menge der zu dokumentierenden Dokumentationsmerkmale) und der

Aufbaustruktur eines Terminologie-Systems waren für die korrekte Verbindung bzw. das korrekte Mapping von Dokumentationsmerkmalen mit Konzepten einer Terminologie/eines kontrollierten Vokabulars entscheidend.

In diesem Abschnitt werden daher Kontrollpunkte definiert, die eine genaue Überprüfung der Architektur bzw. des Modells der jeweiligen eruierten Ansätze hinsichtlich der Umsetzung funktionaler Anforderungen zur terminologischen Fundierung des implementierten Dokumentationssystems ermöglichen. Bei dieser Untersuchung wird der Hauptwert auf die Verbindung zwischen den drei Hauptsichten einer terminologischen Fundierung von Dokumentationssystemen gelegt, nämlich Terminologie, Dokumentation und Datenhaltung. Die Analyse wurde auf die Untersuchung folgender fünf essenzieller Ebenen bzw. Kontrollpunkte mit ggf. Untersuchungsfragen aufgeteilt:

1. Nutzung eines kontrollierten Vokabulars

- Wurde ein kontrolliertes Vokabular angewendet?
- Wie wurde es in dem Modell des Ansatzes eingebunden/integriert und genutzt?

2. Merkmalsebene

- Welche Merkmale, Merkmalsarten und Merkmalsausprägungen wurden dokumentiert?
- Wie wurden sie in dem jeweiligen Ansatz modelliert?
- Wurden sie mit dem angewendeten kontrollierten Vokabular verbunden? Wenn ja, wie?

3. Wertebereiche

- Wurden bzw. wenn ja welche Wertebereiche wurden für die Bereitstellung von Kode-/Wertelisten und Semantiken für Dokumentationseinheiten definiert?
- Wurden kontrollierte Vokabulare für die Definition von Wertebereichen in dem jeweiligen Ansatz benutzt?

4. Rollen und Entitätstypen

Hier soll eine Zusammenfassung aller Objekte und Objekttypen im Modell des Ansatzes aufgeführt werden, die kontrollierte Vokabulare, Merkmalsebenen und Wertebereiche abbilden.

5. Einschränkungen

Hier sollen Einschränkungen des jeweiligen untersuchten Ansatzes erläutert und ggf. diskutiert werden.

## D ANALYSE BESTEHENDER ANSÄTZE

---

### D.1 EINFÜHRUNG

Im vorherigen Kapitel wurden die Anforderungen und Architektur eruiertes Ansätze zur terminologischen Fundierung von Dokumentationssystemen in den letzten Dekaden vorgestellt, um sowohl allgemeine als auch spezifische Informationen über notwendige Merkmale, Merkmalsarten und Merkmalsausprägungen zu bekommen, deren Dokumentation ein medizinisches Dokumentationssystem unterstützen muss. Interessant waren ebenfalls die Untersuchung der praktischen Anwendung und der Nutzen eines kontrollierten Vokabulars in den jeweiligen gefundenen Dokumentationssystemen.

Der Entwurf und die Entwicklung eines eigenen Modells für ein terminologisch fundiertes Dokumentationssystem, sowie die Erarbeitung dessen Anforderungen bauen auf die Ergebnisse der Analyse existierender Ansätze auf, die im Folgenden erläutert werden. Diese Analyse baut auf die im vorherigen Kapitel vorgestellten Analyse- und Qualitätskriterien auf, die eine effektive Untersuchung und Bewertung der Ansätze ermöglichen.

Die Zusammenfassung der Ergebnisse der Untersuchung existierender Modelle wird jeweils beim Top-Down-Ansatz in der Tabelle 1 und beim Bottom-Up-Ansatz in der Tabelle 2 dargelegt. In diesen Tabellen wird die Erfüllung der Qualitätskriterien von dem jeweiligen Ansatz mit folgenden Zeichen gekennzeichnet: das OK-Zeichen kennzeichnet die vollständige Erfüllung des jeweiligen betrachteten Qualitätskriteriums; das Kreuz-Zeichen kennzeichnet, dass das Qualitätskriterium bei dem jeweiligen Ansatz überhaupt nicht erfüllt ist; und das Frage-Zeichen kennzeichnet, dass das Qualitätskriterium bei dem jeweiligen Ansatz nur teilweise erfüllt ist.

### D.2 TOP-DOWN-ANSATZ: ISO/IEC 11179 ED.3

#### D.2.1 Nutzung eines kontrollierten Vokabulars (*terminologische Fundierung*)

Relevante Regionen des ISO/IEC 11179 Metamodells in seiner dritten Version wurden zur Abbildung medizinischer Vokabulare zu Grunde gelegt: *Concept System Metamodel Region* und *Classification Metamodel Region*. Vokabulare können somit durch das ISO-Element *Concept\_System* dargestellt werden und Vokabular-Elemente durch das ISO-Element *Concept*. Beziehungen zwischen *Concepts* innerhalb eines *Concept\_System* werden durch die ISO-Elemente *Link*, *Link\_End* und *Relation\_Role* dargestellt. Für die Abbildung einer Klassifikation in ISO/IEC 11179 Ed. 3 gilt folgendes Beispiel: die ICD-10 Krankheitsklasse “Primär insulinabhängiger Diabetes mellitus [Typ-1 Diabetes] mit Koma” mit dem Code E10.0 ist eine “Primär insulinabhängiger Diabetes mellitus [Typ-1 Diabetes]” mit dem Code E10.-. Beide Krankheitsklassen gehören zum *Concept\_System* “ICD-10-GM 2012”, werden als *Concept* in ISO/IEC 11179 aufgenommen und durch den *Link* “Hierarchy” verknüpft. Wobei dem *Concept* “Primär insulinabhängiger Diabetes mellitus [Typ-1 Diabetes] mit Koma” als *Link\_End* die *Relation\_Role* “is\_a” zugeordnet wird, da er in einer untergeordneten hierarchischen Beziehung zum *Concept* “Primär insulinabhängiger Diabetes mellitus [Typ-1 Diabetes]” steht. Umgekehrt erhält der

*Concept* “Primär insulinabhängiger Diabetes mellitus [Typ-1 Diabetes]” als zweiter *Link\_End* dieser monohierarchischen Beziehung die *Relation\_Role* “inverse\_isa” (s. Abbildung 8).

### **D.2.2 Merkmalsebene**

Um Merkmale einer Dokumentationseinheit bzw. um Datenelemente zu beschreiben wird auf das *Data Description Package* des ISO/IEC 11179 zurückgegriffen. Die vereinfachte Version dieses Metamodells zeigt, wie ein Datenelement modelliert werden kann (s. Abbildung 9). In diesem vereinfachten Datenmodell wird zwischen semantischer Ebene (oberer Teil) und Darstellungsebene (unterer Teil) bei der Definition von Merkmalen unterschieden. Angenommen, man möchte auf ein elektronisches Case Report Form (CRF), welches ein Dokumentationssystem darstellt, den Namen einer krebshemmenden Substanz eingeben, bildet folgendes Szenario eine terminologische Fundierung des eCRFs ab. Das entsprechende *Data\_Element* für den Namen einer krebshemmenden Substanz soll zuerst im *Data Description Package* erzeugt werden. Dies erfolgt in der semantischen Ebene mit der Identifizierung aller möglichen Medikamente und Chemikalien im *Conceptual\_Domain* und mit der Bildung des zu diesem *Data\_Element* zugehörigen Konzepts im *Data\_Element\_Concept*. In der Darstellungsebene wird dann ableitend von dem definierten Konzept das eigentliche Dokumentationsmerkmal erzeugt, und ihm ein möglicher Wert aus der vorab festgelegten *Value\_Domain* zugewiesen. Die *Value\_Domain* definiert den Wertebereich und umfasst mögliche Werte für den Namen einer krebshemmenden Substanz, z.B. {“Ursodiol”, “Grüner Tee”, ...} (s. Abbildung 9). Terminologien und Klassifikationen spielen dabei eine wichtige Rolle, da sie sowohl Semantiken (z.B. *Value\_Meaning*) für die erzeugten Konzepte als auch Wertelisten/Wertebereiche (*Value\_Domain*) für die definierten Merkmale/Datenelementen bereitstellen. Somit wird die Standardisierung von den Merkmalen/Datenelementen gewährleistet. Nach Spezifikation des Dokumentationsmerkmals könnte die eCRF-Sicht auf

Abbildung 10 entstehen: das *Data\_Element* wird durch die Frage „Welche krebshemmende Substanz nehmen Sie?“ dargestellt und die *Permissible\_Values* aus einem kontrollierten Vokabular werden als Auswahlliste angeboten.

### **D.2.3 Wertebereiche**

Wertebereiche für Merkmale/Datenelemente werden durch *Value\_Domain* definiert, die aus *Permissible\_Values* bestehen kann. Für die Definition von Wertebereichen eignen sich hier kontrollierte Vokabulare; deren Nutzung ist allerdings auf eine flache Werteliste beschränkt.

### **D.2.4 Rollen und Entitätstypen**

Jede Klasse oder jeder Begriff eines kontrollierten Vokabulars ist ein *Concept* mit folgenden Rollen: *Object\_Class*, *Characteristic*, *Data\_Element\_Concept*, *Conceptual\_Domain*, *Value\_Meaning*, *Relation*, *Relation\_Role*, *Unit\_of\_Measure*. Alle Elemente des ISO/IEC 11179-Metamodells können *Designatable\_Items* (d.h. besitzen eine Bezeichnung) und/oder *Classifiable\_Items* (d.h. können

klassifiziert werden) sein. *Classifiable\_Items* können durch Konzepte (*Concept*) klassifiziert werden. Ein *Concept\_System* besteht aus *Concepts*, die mit *Links* verbunden werden und bestimmte *Relation\_Roles* haben können.

Auf Merkmalsebene werden *Data\_Element* und *Value\_Domain* benutzt, um Dokumentationsmerkmale und ihre Ausprägungen (Wertebereich) zu beschreiben.

### **D.2.5 Einschränkungen**

Im Allgemeinen ist die Abbildung von Klassifikationen als Werteliste in *Concepts Package* des ISO/IEC 11179 einfach. Abbildung 9 zeigt die mögliche Nutzung von kontrollierten Vokabularen aus dem *Concepts Package* für die Bereitstellung sowohl von Semantiken für Konzepte aus der semantischen Ebene (z.B. *Conceptual\_Domain*, *Value\_Meaning*) als auch von Wertelisten (*Value\_Domain*) für Datenelemente (*Data\_Element*). Dies stellt allerdings ein „nice-to-have“ bzw. einen Untersuchungspunkt mit dem ISO/IEC 11179 Metamodell für die terminologische Fundierung eines Dokumentationssystems dar, da die korrekte Verbindung zwischen Elementen des *Data Description Package* und Elementen des *Concepts Package* fehlt. Beim Versuch ist die Redundanz der Elemente eines kontrollierten Vokabulars unvermeidbar. Man sollte z.B. Konzepte eines Vokabulars sowohl in der *Enumerated\_Value\_Domain* als Werteliste für Items aufnehmen, als auch *Concepts* eines *Concept\_System*. Nimmt man aber Elemente eines kontrollierten Vokabulars, für deren Nutzung als Werteliste für Items, lediglich in die *Enumerated\_Value\_Domain* auf (ohne eine vorherige Abbildung im *Concepts Package*), so gehen die Struktur der Hierarchie und die Semantik dieser Elemente verloren. Die bessere Variante wäre, Konzepte eines Vokabulars als *Value\_Meaning* für die Definition von Items/Datenelementen aufzunehmen. Dadurch, dass *Value\_Meaning* Konzepte (*Concept*) sind, besteht hier keine Gefahr der Redundanz der Einträge. Schließlich könnte für die Verknüpfung von Merkmalsdefinitionen mit kontrollierten Vokabularen im ISO/IEC 11179 an drei Möglichkeiten gedacht werden: die Nutzung eines kontrollierten Vokabulars als Wertebereich bei deren vollständigem Import ins System, den Verweis auf das kontrollierte Vokabular über einen Object Identifier (OID), sowie die Kopplung mit einem Terminologie-Server. ISO bietet allerdings keine Struktur, um auf Vokabulare mittels ihrer OIDs im *Permissible\_Value* zu verweisen.

Schließlich zeigt ISO/IEC 11179 wichtige Einschränkungen bezüglich der Abbildung von komplexen Terminologien wie SNOMED CT. Eine Erweiterung des Standards, die *Links* als Instanzen von *Concepts* definieren würde, würde zur korrekten Transformation von SNOMED CT in ISO/IEC 11179 beitragen. Die Abbildung von Versionierungen und Sprachvarianten von Vokabularen sind noch zu untersuchen [6].

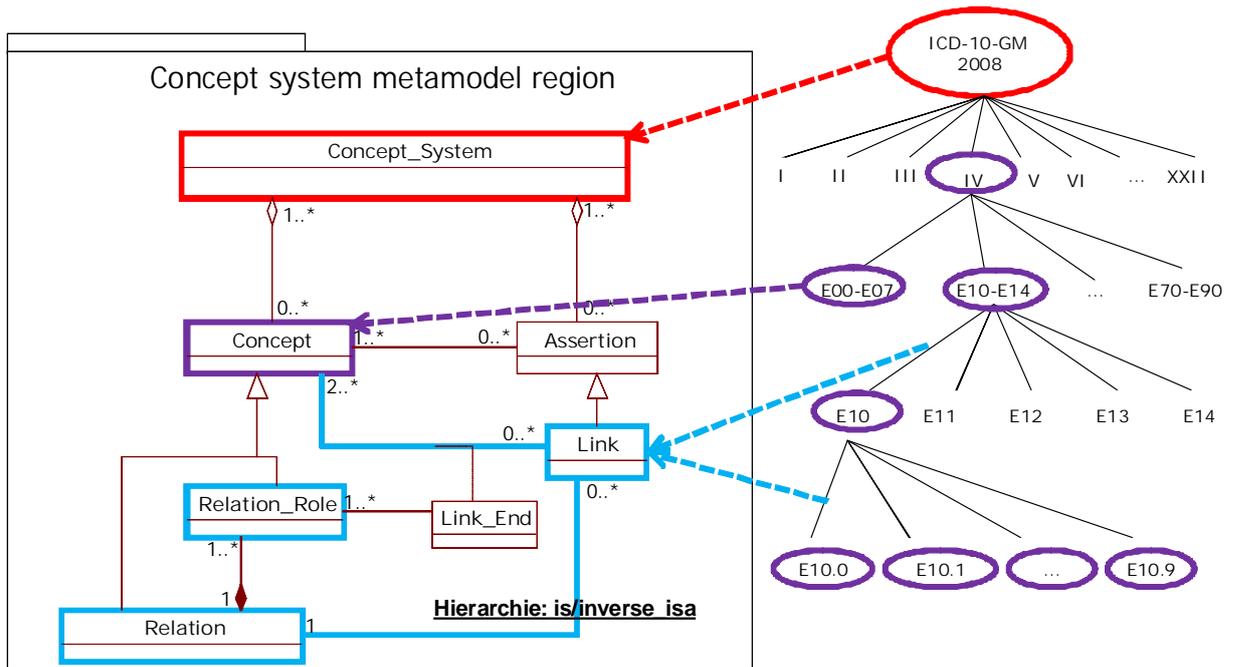
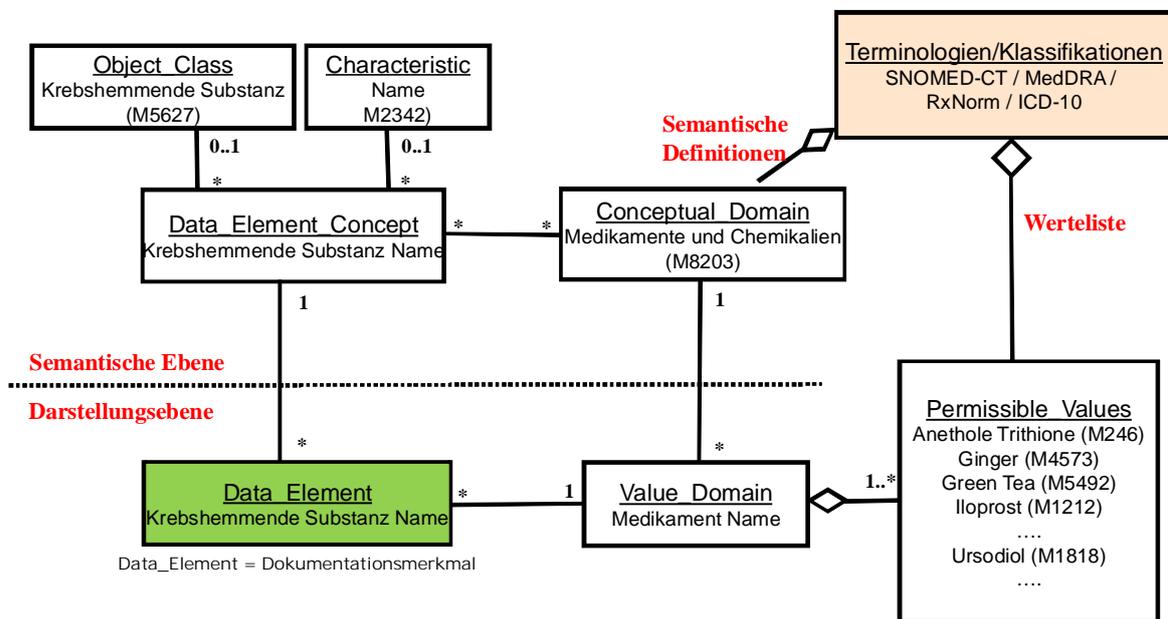


Abbildung 8: Einbettungsmöglichkeit von ICD-10-GM in die *Concept System Metamodel Region* des ISO/IEC 11179 Ed.3 *Concepts Package*



**High-level Data Description Metamodel: Abbildung von Dokumentationsmerkmalen**

Abbildung 9: Erzeugung eines Dokumentationsmerkmals im *Data Description Package* des ISO/IEC 11179 Ed.3 und Nutzungsmöglichkeit von kontrollierten Vokabularen auf semantische und Darstellungsebene

The image shows a screenshot of an electronic Case Report Form (eCRF) interface. It is divided into sections: 'Angaben zur Person' and 'Krankheiten'. The 'Krankheiten' section contains a question: '2. Welche der folgenden Krankheiten hatten Sie jemals?' with options for 'Bluthochdruck, Hypertonie' and 'Venenthrombose', each with 'Ja' and 'Nein' radio buttons. Below this is question '3. Welche Krebshemmende Substanz nehmen Sie bis jetzt?' with a dropdown menu. The dropdown menu is open, showing a list of substances: Anethole Trithione, Anethole Trithione, Ginger, Green Tea (highlighted), Ilprost, Ursodiol, and an ellipsis. A black oval encircles the question and the dropdown menu. An arrow labeled 'Data\_Element' points to the question text, and another arrow labeled 'Permissible\_Values' points to the dropdown list.

**Abbildung 10: Darstellungsmöglichkeit eines *Data\_Element* auf einem elektronischen Case Report Form (eCRF). Wertelisten werden nicht manuell eingegeben sondern direkt aus einem kontrollierten Vokabular übernommen.**

Tabelle 1 zeigt die Zusammenfassung der Ergebnisse der Untersuchung existierender Modelle beim Top-Down-Ansatz.

**Tabelle 1: Erfüllungsstatus der Prüfkriterien vom ISO/IEC 11179 Ed.3 für eine terminologische Fundierung.**

	Vokabular	Rollen	Merkmalswelt	Speicher für Wertebereiche
ISO/IEC 11179 Ed.3	✓	✓	?	✓

### D.3 BOTTOM-UP-ANSATZ

#### D.3.1 *TERM*Trial

##### D.3.1.1 Nutzung eines kontrollierten Vokabulars (terminologische Fundierung)

*TERM*Trial setzt in seinem Datenmodell die Nutzung eines konzeptorientierten Terminologie-Managementsystems (mit Anwendung einer Referenzterminologie bzw. einer studienspezifischen Terminologie) als Grundlage für das Dokumentationssystem voraus. Im Terminologie-Managementsystem (*TMS*) von *TERM*Trial können Konzepte (*Concepts*) aus Vokabularen verwendet werden. Diese *Concepts* können mehreren Bezeichnungen bzw. Synonymen (*Terms*) zugewiesen werden, stellen Semantiken und Bezeichnungen für Studienmerkmale (*Contexts*) und -merkmalsarten (*Trial Specific Items*), sowie Wertelisten für bestimmte Studienmerkmalsarten (*set of options*) bereit. Explizit nennt Merzweiler in seiner Dissertation keine spezifische Referenzterminologie. *TERM*Trial ermöglicht nichtsdestotrotz die Bereitstellung von integrierten standardisierten Terminologie-Systemen, aus denen standardisierte Dokumentationssysteme konsistent abgeleitet werden können. Zudem ermöglicht *TERM*Trial die Nutzung gleicher *Trial Specific Items* in unterschiedlichen klinischen Studien, sprich in unterschiedlichen eCRFs, was die Harmonisierung (Vereinheitlichung und Vergleichbarkeit) und Standardisierung von Studienitems unterstützt.

##### D.3.1.2 Merkmalsebene

In der Analysephase zur Entwicklung von *TERM*Trial wurden Strukturen und Inhalte von Studienformularen der deutschen Gesellschaft für Pädiatrische Onkologie und Hämatologie (GPOH) untersucht. So wurden Informationen über relevante Dokumentationseinheiten (*object types* und deren Datentypen) gesammelt. Die hier erfassten Beispielmerkmale betreffen also überwiegend die Domain der Onkologie und Hämatologie.

Auf Merkmalsebene werden lediglich *Concepts* mit den Rollen *Context* und *Trial Specific Item*, wie sie in dem *Trial Specific Terminology Module* definiert wurden, in einer monohierarchischen Baumstruktur für die Erzeugung des eCRF angezeigt (s. Abbildung 3).

##### D.3.1.3 Wertebereiche

Die möglichen Werte bzw. Merkmalsausprägungen einer Merkmalsart (*Trial Specific Item*) werden als *set of option* definiert. Sie werden zwar bei der Erzeugung eines eCRF mit dem *Form Building Component* nicht angezeigt, ein Mapping zwischen beiden Entitäten (*Trial Specific Item* und zugehörigem *set of option*) existiert jedoch in der generierten Studiendatenbank. Diese Werteliste wird im erstellten Dokumentationssystem bei der Erfassung oder bei der Suche nach einem *Concept* angezeigt. Wenn man sich an das gegebene Beispiel zur Dokumentation der Relevanz einer Erkrankung anlehnt, müssen Elemente einer *set of option* nicht zwangsläufig oder explizit aus einem kontrollierten Vokabular stammen. Merzweiler weist lediglich darauf hin, dass jede *option* durch mehrere *Concepts* abgebildet werden kann.

#### D.3.1.4 Rollen und Entitätstypen

Jeder Begriff einer Terminologie ist ein *Concept* mit Synonymen (*Terms*) und folgenden Rollen: *Object type* bzw. *Context*, *Item* bzw. *Trial Specific Item*. Eine klinische Studie wird durch das Modellelement *Clinical trial* definiert, das sich aus hierarchischen strukturierten *Contexts* und *Trial Specific Items* zusammensetzt.

#### D.3.1.5 Einschränkungen

Konzepte in TERMTrial dienen zwar der Bezeichnung und Definition von Merkmalen bzw. Studienitems, allerdings besteht die Werteliste (*set of options*) für diese Studienitems nicht unbedingt direkt aus *Concepts* einer Terminologie, sondern jede *Option* dieser Werteliste kann durch mehrere *Concepts* abgebildet bzw. mehreren *Concepts* zugeordnet werden. Dies könnte eine Verdoppelung der Einträge bedeuten.

Der Prozess des Formularaufbaus erfolgt unter strenger terminologischer Steuerung, so dass lediglich in der Baumstruktur angezeigte *Contexts* und *Items/Trial Specific Item* benutzt werden können. Falls andere oder neue *Contexts* und *Items/Trial Specific Items* für eine Studie benötigt werden, müssen sie vorab im *Terminology Management System (TMS)* definiert werden. Die Modellierung eines neuen eCRF für eine Studie würde also in diesem Fall die Generierung eines neuen Studiendatenmodells bzw. die Erzeugung einer neuen Studiendatenbank bedeuten.

### D.3.2 InCoMe

#### D.3.2.1 Nutzung eines kontrollierten Vokabulars (terminologische Fundierung)

In InCoMe können kontrollierte Vokabulare nicht explizit integriert und benutzt werden, um „Stammdaten“ zu definieren, da Strukturen diesbezüglich fehlen. Sämtliche „Stammdaten“, die für die Definition des Wertebereiches bestimmter Merkmalsarten benötigt werden, müssen im CDMS-Designer manuell erfasst werden.

#### D.3.2.2 Merkmalsebene

Das Dokumentationsmodell von InCoMe besteht aus drei wichtigen Modellelementen für die Bildung der Merkmalsebene: „Merkmalsarten“, „Stammdaten“ und „Kontexte“. Zu Testzwecken wurden Merkmale aus dem HIVNET und HepNet ausgewählt und in InCoMe erfasst. Prinzipiell lassen sich aber in InCoMe beliebig viele Merkmale aus unterschiedlichen medizinischen Bereichen erfassen.

In InCoMe ist die Verknüpfung zwischen Merkmalswelt und kontrollierten Vokabularen nicht vorgesehen.

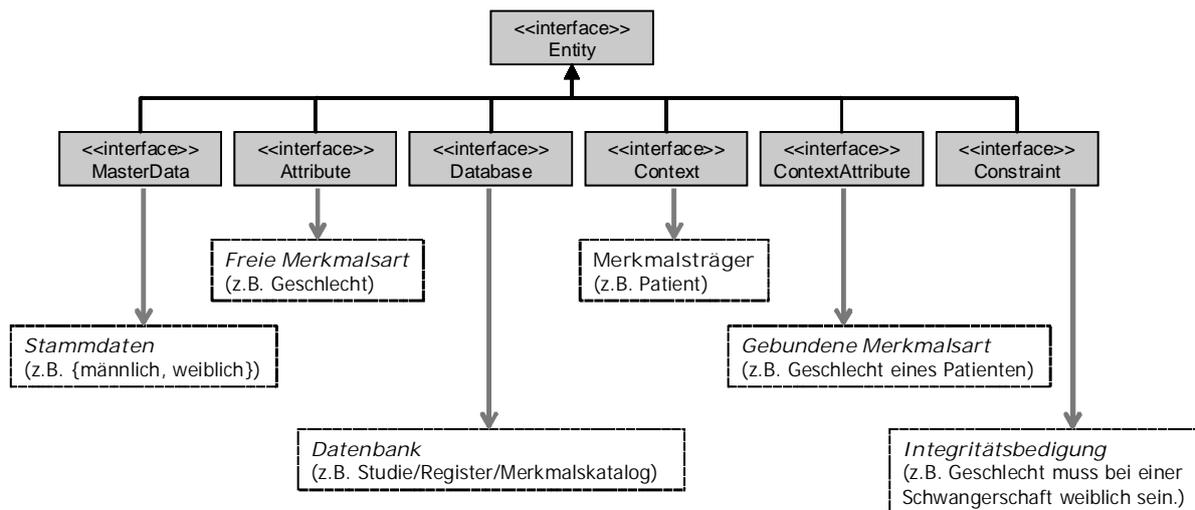
#### D.3.2.3 Wertebereiche

Distinkte Wertebereiche für „Merkmalsarten“ werden durch „Stammdaten“ definiert. Die Einschränkung der Merkmalsausprägungen durch „Stammdaten“ bei der Modellierung einer „Merkmalsart“ ist allerdings optional. Mögliche Werte für Merkmalsarten können nämlich auch

Ganzzahlen, Dezimalzahlen, usw. oder vom Typ Datum (z.B. für Geburtsdatum) sein. Die Verknüpfung von Merkmalsausprägungen bzw. Wertebereiche mit kontrollierten Vokabularen fehlt in InCoMe. Alle „Stammdaten“ müssen jedes Mal manuell erfasst werden.

#### D.3.2.4 Rollen und Entitätstypen

Jedes Modellelement des Dokumentationsmodells von InCoMe ist eine „Entität“ mit folgenden Rollen: „freie Merkmalsart“, „gebundene Merkmalsart“, „Stammdaten“, „Kontext“, „Integritätsbedingung“, „Datenbank“, „Applikation“, wobei eine „gebundene Merkmalsart“ eine „freie Merkmalsart“ ist, der ein „Kontext“ zugeordnet ist (s. Abbildung 11 für Beispiele).



**Abbildung 11: Modellelemente mit Beispielen zur Dokumentation von Informationen im Dokumentationssystem (CDMS) von InCoMe**

#### D.3.2.5 Einschränkungen

Eine terminologische Fundierung des von InCoMe generierten Dokumentationssystems existiert nicht. Für die Erfassung (Bezeichnung und Beschreibung) von freien Merkmalsarten, sowie für die Definition ihrer Wertebereiche wird weder auf Terminologien noch auf Klassifikationen zurückgegriffen. Das Dokumentationssystem von InCoMe wird vom CDMS-Designer aus durch den Ansatz von Plugins generiert. Somit erzeugt InCoMe ein standalone Application Programming Interface (API, Programmierschnittstelle) für das Clinical Data Management System (CDMS). Dies hat zu Folge, dass bei Änderungen im CDMS-Designer das CDMS auch neu generiert werden muss.

### D.3.3 ORCA

#### D.3.3.1 Nutzung eines kontrollierten Vokabulars (terminologische Fundierung)

In ORCA kommen die dargestellten medizinischen Konzepte lediglich aus einem vordefinierten Thesaurus. Dieser Thesaurus besteht aus allen möglichen definierten medizinischen Begriffen bzw. Bezeichnungen. Dank diesem Thesaurus entsteht die Wissensbasis von ORCA, wodurch das Knowledge Model des Systems entworfen werden kann. Es besteht allerdings keine Verknüpfung

zwischen diesem Thesaurus und Standards bzw. kontrollierten Vokabularen, um den Thesaurus zu standardisieren bzw. dessen Inhalt zu harmonisieren. Da der Thesaurus von ORCA möglichst umfangreich sein sollte und ORCA u.a. eine vernetzte Versorgung fördern soll, würde die Einbindung von kontrollierten Vokabularen sehr hilfreich sein und zugleich die semantische Interoperabilität zwischen beteiligten Versorgungszentren unterstützen.

#### D.3.3.2 Merkmalsebene

Auf Merkmalsebene wird bei ORCA auf das entworfene Knowledge Base (Domain Model) zugegriffen, welches die Auswahl der zu dokumentierenden Merkmale ermöglicht und diese funktional in einem konzeptuellen Graph (Baum oder Netzwerk) organisiert. Die Auswahl von Merkmalen aus dem Knowledge Base für das Dokumentationssystem hängt von dem medizinischen Gebiet ab (deshalb spricht man auch vom „Domain Model“). Pro medizinisches Gebiet wird ein entsprechendes Modell mit Hilfe der Knowledge Base entworfen. Dabei spielt natürlich der zugrunde gelegte Thesaurus eine wichtige Rolle, da Merkmale bzw. medizinische Begriffe aus möglichst unterschiedlichen medizinischen Gebieten darin enthalten sind. Ein medizinisches *Concept* aus dem Thesaurus kann als *descriptor* eines anderen *Concepts* in einem bestimmten Kontext oder als *option* oder *unit* benutzt werden.

#### D.3.3.3 Wertebereiche

In ORCA existieren keine expliziten Wertebereiche für Merkmale oder Konzepte aus dem Thesaurus. Auswahllisten werden als einzelner Begriff im Thesaurus mitgespeichert (z.B. ja\_nein, ja\_nein\_unbekannt, etc.) und erst bei der Generierung des Dokumentationssystems bereitgestellt.

#### D.3.3.4 Rollen und Entitätstypen

Jedes medizinische Konzept im ORCA-Thesaurus kann abhängig von seinem Anwendungskontext und seinem Beziehungstyp zu anderen medizinischen Begriffen folgende Rollen haben: *feature*, *descriptor*, *option* and *unit*. Nicht alle Rollen sind allerdings explizit Bestandteile des *Domain Models* von ORCA, sondern manche lassen sich implizit aus den sechs Beziehungstypen ableiten.

#### D.3.3.5 Einschränkungen

Beziehungen zwischen Konzepten gehen im Thesaurus verloren. Die Semantik der Konzepte entsteht lediglich durch den Kontext, der durch den Aufbau des Graphs in der Knowledge Base (Baums oder Netzwerks) gebildet wird.

Selbst wenn medizinische kontrollierte Vokabulare eine logische Basis darstellen, um SDE zu unterstützen, bemängelt Ginneken, dass sie die notwendigen Informationen zur Generierung intuitiver Benutzeroberflächen nicht darstellen, um der Eingabe relevanter kontext-spezifischer Daten vorzugreifen. Sie erkennt allerdings, dass Standard-Terminologien nichtsdestotrotz die Einheitlichkeit von Metadaten für die Dateneingabe verbessern können und das semantische Wiederauffinden von Daten ermöglichen können. Sie schlägt deswegen ein Mapping von bereits existierenden

medizinischen Konzepten aus der Knowledge Base zu kontrollierten medizinischen Vokabularen/Terminologien, um Vorteile von beiden Strukturen auszunutzen [19].

#### **D.3.4 ICNP-based ENRS**

##### D.3.4.1 Nutzung eines kontrollierten Vokabulars (terminologische Fundierung)

Für die terminologische Fundierung des ENRS der Seoulschen Universitätsklinikum wurde die International Classification of Nursing Practice (ICNP) zugrunde gelegt und in der Form von präkoordinierten Pflegedatensätzen benutzt. Diese präkoordinierten Pflegedatensätze werden vom Pflegepersonal für die Dateneingabe benutzt. Der Pflegebegriffsmanager (Nursing Term Manager) ist für die Erzeugung, Aktualisierung, Pflege und Wartung der Pflegeterminologie einschließlich ICNP-Konzepten und kontrollierten präkoordinierten Pflegedatensätzen zuständig.

##### D.3.4.2 Merkmalsebene

Merkmale im ICNP-based ENRS bilden Pflegeaktivitäten und die zu dokumentierenden klinischen Informationen des Seoulschen Universitätsklinikums Bundang ab.

Auf Merkmalsebene werden lediglich präkoordinierte Pflegedatensätze zur Auswahl für die Erfassung von Patientendaten angezeigt. Jeder präkoordinierte Pflegedatensatz kann nämlich einem Dokumentationsmerkmal auf dem CRF zugeordnet werden. Dieser präkoordinierte Pflegedatensatz wird auf Datenbankebene durch mindestens ein Konzept aus der ICNP-Terminologie abgebildet. Im Pflegeterminologie-Management System können zudem optional ICNP-Attribute festgelegt werden, um die Granularität präkoordinierter Pflegedatensätze zu erhöhen. ICNP-Attribute sind Eigenschaften von ICNP-Konzepten (s. Anhang K.1), die bei der Auswahl des zugehörigen präkoordinierten Pflegedatensatzes im ENRS auch selektiert und erfasst werden müssen. Das Konzept „Bewusstsein“ könnte zum Beispiel den „Level“ des Bewusstseins als Attribut haben. Bei der Auswahl des präkoordinierten Pflegedatensatzes „Observe Consciousness“ (Beobachtetes Bewusstsein) bei der Erfassung eines Patientendatensatzes im ENRS wird dem Pflegepersonal die zu diesem Pflegedatensatz zugeordnete Attributen-Liste zur Werteeingabe angezeigt (s. Abbildung 12).

Intensive Care Units	Admission Nursing Assessment
Medical Intensive Care Unit	Pat. ID: <input type="text" value="88888888"/>
Neonatal Intensive Care Unit	Surname: <input type="text" value="Surname"/>
Surgical Intensive Care Unit	First Name: <input type="text" value="Firstname"/>
<b>Respiratory Intensive Care Unit</b>	<b><u>Physical Assessment</u></b>
<ul style="list-style-type: none"> <li>- Body Temperature</li> <li>- <b>Observe Consciousness</b></li> <li>- Systolic Blood Pressure</li> <li>- Diastolic Blood Pressure</li> <li>- ...</li> </ul>	Item: <input type="text" value="consciousness"/>
	Level: <input type="text" value="Low"/>
	<b><u>Admission Information</u></b>
	Date: <input type="text" value="dd.mm.yyyy"/>

**Abbildung 12: Beispiel der Erfassung des „Level“ des „Observe Consciousness“ eines Patienten im ENRS**

#### D.3.4.3 Wertebereiche

Im ENRS werden ICNP Konzepte präkoordiniert und auf Merkmalebene benutzt, um standardisierte Dokumentationseinheiten im Pflegesystem einzugeben. Sie werden allerdings nicht verwendet, um Wertebereiche zu definieren. Werte für ICNP Attribute sind auf der Anwendungsebene manuell einzugeben.

#### D.3.4.4 Rollen und Entitätstypen

Jedes Element der ICNP Terminologie ist ein Konzept (*Concept*) mit optionalen Attributen (*Characteristic*). Diese Konzepte werden semantisch zusammengesetzt, um neue Konzepte zu erzeugen, sogenannte präkoordinierte Pflegedatensätze (*precoordinated statements*). Die präkoordinierten Pflegedatensätze bilden die wichtigsten Dokumentationseinheiten des ENRS.

#### D.3.4.5 Einschränkungen

In dem vorgestellten ENRS wird lediglich auf die ICNP Terminologie zurückgegriffen und nur präkoordinierte Pflegedatensätze aus ICNP-Konzepten finden Anwendung auf Merkmalebene des Dokumentationssystems. Man könnte hier an die Einbindung und Nutzung weiterer, für die

Dokumentation von Pflegedaten passenden, Terminologien denken, um die terminologische Fundierung des ENRS zu verbessern.

#### **D.4 ZUSAMMENFASSUNG DER ERGEBNISSE**

Aus Tabelle 1 und Tabelle 2 lassen sich folgende Ergebnisse aus der Analyse bestehender Ansätze zur terminologischen Fundierung von Dokumentationssystemen zusammenfassen:

Der ISO/IEC 11179 Standard stellt zwei wichtigen Datenmodelle zur möglichen Modellierung eines terminologisch fundierten Dokumentationssystems dar: Das *Data Description Package* und das *Concepts Package*. Allerdings wurden bei seiner Untersuchung Unklarheiten festgestellt, die erst erhoben werden müssen, damit der Standard sich reibungslos als Metamodell für die terminologische Fundierung von Dokumentationssystemen eignet: in der *Enumerated\_Value\_Domain* ist die Darstellung von kontrollierten Vokabularen auf eine flache Struktur von Begrifflisten beschränkt, so dass sämtliche Hierarchien zwischen Elementen einer Klassifikation verloren gehen; während zudem die meisten Terminologien nur eine Liste von Begriffen darstellen, müssen bei Klassifikationen komplizierte Regelwerke zur richtigen Benutzung von Klassen miteinbezogen und abgebildet werden. Klassifikationsregeln lassen sich jedoch schwer in *Data Description Package* abbilden, wegen der erforderlichen flachen Struktur im *Enumerated\_Value\_Domain*. Wünschenswert beim ISO/IEC 11179 Standard ist schließlich die Nutzung von kontrollierten Vokabularen sowohl im *Concepts Package* (als *Concepts*) als auch im *Data Description Package* als Wertebereiche für Items, ohne Redundanzen der Elemente eines kontrollierten Vokabulars. Somit erfüllt der ISO/IEC 11179 Standard nur bedingt die definierten Analyse Kriterien für die Modellierung eines terminologisch fundierten Dokumentationssystems.

TERMTrial stellt ein sehr gut durchdachtes Terminologie-basiertes Dokumentationssystem für klinische Studien dar. Das entwickelte Terminologie-Managementsystem ermöglicht die Bereitstellung von integrierten standardisierten Terminologie-basierten Merkmalsausprägungen, aus denen standardisierte Dokumentationssysteme einheitlich abgeleitet werden können. Es ermöglicht ebenfalls die Wiederverwendung von Studienitems in unterschiedlichen klinischen Studien zur Verbesserung der Harmonisierung und Standardisierung von Studienitems. In der praktischen Anwendung gibt es jedoch keinen Hinweis auf die Verwendung von standardisierten kontrollierten Vokabularen, wie ICD-10, MedDRA, LOINC, SNOMED-CT, etc. Das *Terminology Management System* (TMS) verwaltet medizinische und studienspezifische Begriffe, die höchstwahrscheinlich in Studien der GPOH verwendet wurden. TERMTrial erfüllt somit nur teilweise die festgelegten Analyse Kriterien. Der Arbeitsaufwand bei einer erforderlichen neuen Generierung einer Studiendatenbank bei Änderungen im *Terminology Management System* lässt sich zudem kaum vermeiden.

InCoMe sieht die Einbindung von kontrollierten Vokabularen für den terminologischen Support bei der Modellierung von Dokumentationssystemen im CDMS-Designer nicht vor. Kontrollierte

Vokabulare sind weder für die Definition von Merkmalen noch für die Definition von Wertelisten eingesetzt. InCoMe erfüllt dadurch die wichtigste Anforderung für die Modellierung eines terminologisch fundierten Dokumentationssystems nicht. Es stellt allerdings ein attraktives Werkzeug für die Modellierung von Plattform- und Datenbankmanagementsystem-unabhängigen Dokumentationssystemen. Der Mehrwert in InCoMe ist außerdem sicherlich die Berücksichtigung von Integritätsbedingungen zur Gewährleistung von Integrität, Gültigkeit und Richtigkeit der eingegebenen Daten.

ORCA fördert die strukturierte Dateneingabe in einem Dokumentationssystem für die Forschung und Versorgung, sowie für eine schnelle Datenabfrage. Das Modell von ORCA ermöglicht auch die Implementierung verschiedener medizinischer Domänen ohne Einschränkungen auf das Detailniveau. Da jedoch der Thesaurus in ORCA lediglich eine Sammlung von medizinischen Begriffen ist, werden die semantischen Beziehungen zwischen den Konzepten durch das Domain Model bzw. den medizinischen Kontext (z.B. Radiologie, Neurologie, Pathologie, etc.) hergestellt. Dieser Kontext ist wiederum durch einen konzeptuellen Graph in der Knowledge Base definiert. Wie Van Ginneken bereits erkannte: Selbst wenn die Integration von kontrollierten medizinischen Terminologien in die Knowledge Base nicht einfach ist, können kontrollierte Vokabulare verwendet werden, um die Einheitlichkeit der Metadaten für die Dateneingabe zu verbessern und ein besseres Wiederauffinden von Daten zu ermöglichen. Als praktikable Strategie wurde ein Mapping von medizinischen Begriffen aus dem im ORCA verwendeten Thesaurus zu Konzepten einer kontrollierten medizinischen Terminologie empfohlen, um eine effektive und effiziente terminologische Fundierung zu erzielen (Ginneken 2003). ORCA stellt zudem keinen expliziten Terminologie-basierten Speicher für Wertebereiche von Dokumentationsmerkmalen zur Verfügung. Somit erfüllt ORCA die relevanten Anforderungen für ein Terminologie-basiertes Dokumentationssystem nicht.

Während das Seoulsche Electronic Nursing Record System eine gute Grundlage zur Untersuchung bzw. Analyse wichtiger Faktoren bei der Entwicklung eines Terminologie-basierten Pflegedokumentationssystems bietet, basiert es lediglich auf ICNP bzw. auf der Nutzung von ICNP-basierten präkoordinierten Pflegedatensätzen im Dokumentationssystem. Zudem existieren keine Terminologie- bzw. ICNP-basierten Wertebereiche für die vordefinierten Dokumentationsmerkmale. Somit erfüllt das ENRS die definierten Analyse Kriterien für eine terminologische Fundierung von Dokumentationssystemen nur teilweise.

Nach dieser Analyse von existierenden Ansätzen von Terminologie-basierten Dokumentationsmerkmalen und basierend auf den Ergebnissen der Analyse, sowie auf den festgelegten Analyse Kriterien, widmen sich nun die nachfolgenden Kapitel der Entwicklung eines eigenen Modells zur terminologischen Fundierung von Dokumentationssystemen.

Tabelle 2 zeigt die Zusammenfassung der Ergebnisse der Untersuchung existierender Modelle beim Bottom-Up Ansatz.

**Tabelle 2: Zusammenfassung der Erfüllung der Prüfkriterien beim Bottom-Up-Ansatz.**

	<b>Vokabular</b>	<b>Rollen</b>	<b>Merkmalswelt</b>	<b>Speicher für die Wertebereiche</b>
TERMtrial	<b>?</b>	✓	✓	✓
InCoMe	<b>X</b>	✓	✓	✓
ORCA	<b>?</b>	✓	✓	<b>X</b>
ENRS	<b>?</b>	✓	✓	<b>?</b>

## **E MODELLENTWICKLUNG**

---

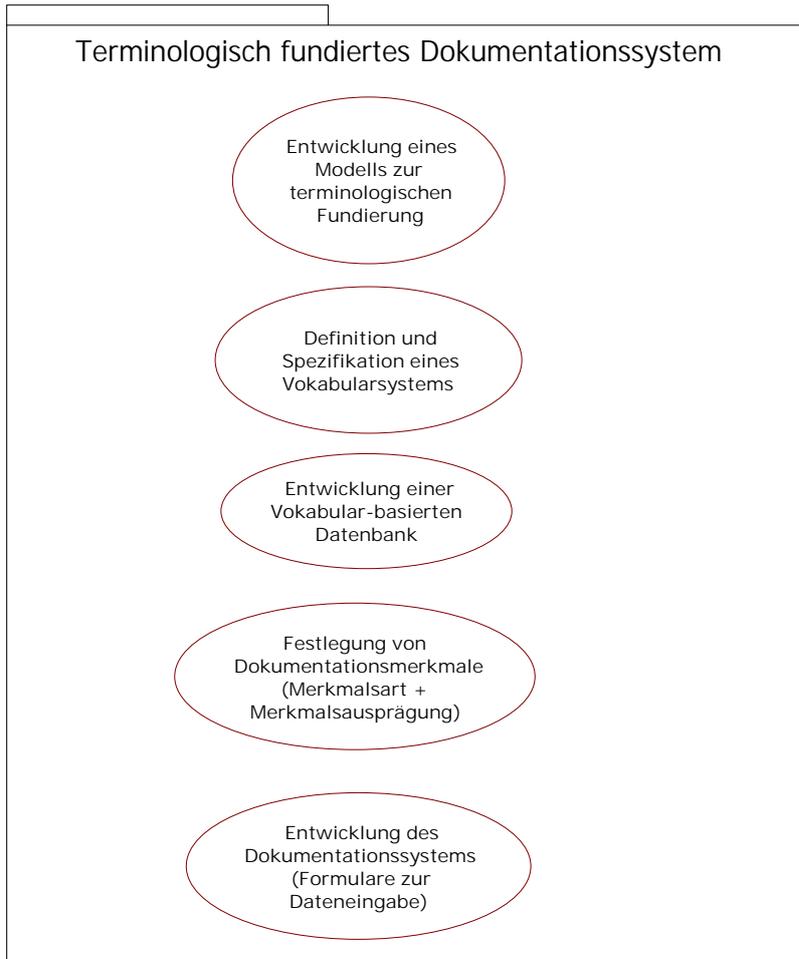
### **E.1 GRUNDIDEE**

Der Abschnitt Modellentwicklung widmet sich dem Entwurf eines eigenen Modells zur terminologischen Fundierung eines Dokumentationssystems namens VokaDoks (Vokabular-basiertes medizinisches Dokumentationssystem).

Aus terminologischer Sicht empfiehlt sich im Allgemeinen drei zentrale Aufgaben bei der Entwicklung von Terminologie-basierten Dokumentationssystemen zu berücksichtigen: die Entwicklung eines Terminologischen Systems (Terminologie), die Entwicklung von Datenbanken (Datenhaltung) und die Entwicklung eines Dokumentationssystems (Dokumentation), das auf das terminologische System zugreifen soll. Wie es sich außerdem der Analyse bestehender Ansätze zur terminologischen Fundierung von Dokumentationssystemen entnehmen lässt, ergeben sich auch für die vorliegende Arbeit drei wesentliche Aufgaben, die bei der Entwicklung eines Modells zur terminologischen Fundierung eines Dokumentationssystems zu berücksichtigen sind:

1. Die Definition und Spezifikation eines Vokabular-Systems sowie die darin enthaltenen kontrollierten Vokabulare (Klassifikationen und Terminologien),
2. Die Festlegung von Dokumentationsmerkmalen mit ihren entsprechenden Merkmalsarten und -ausprägungen, die in dem entwickelten Dokumentationssystem dokumentiert werden müssen, sowie die Beschreibung von Zusammenhängen zwischen diesen Dokumentationsmerkmalen und Konzepten/Begriffen aus den festgelegten kontrollierten Vokabularen bzw. die Beschreibung der Nutzung von kontrollierten Vokabularen zur Bereitstellung von Bezeichnungen oder Semantiken für die Dokumentationsmerkmale. Aus Dokumentationssicht besteht außerdem eine wichtige Aufgabe in der Entwicklung von Formularen zur Dateneingabe (eCRFs) im Dokumentationssystem.
3. Die Entwicklung einer Vokabular-basierten Datenbank, die das im ersten Schritt spezifizierte Vokabular-System zugrunde legt, und festgelegte Dokumentationsmerkmale konsistent speichert.

Aus diesen Aufgaben lassen sich die in Abbildung 13 dargestellten Hauptanwendungsfälle (Use Cases von UML) ableiten, die beim Design eines Dokumentationssystems mit terminologischer Fundierung zu berücksichtigen sind.



**Abbildung 13: UML-Hauptanwendungsfälle als Arbeitsschritte/-pakete zur Entwicklung eines terminologisch fundierten Dokumentationssystems**

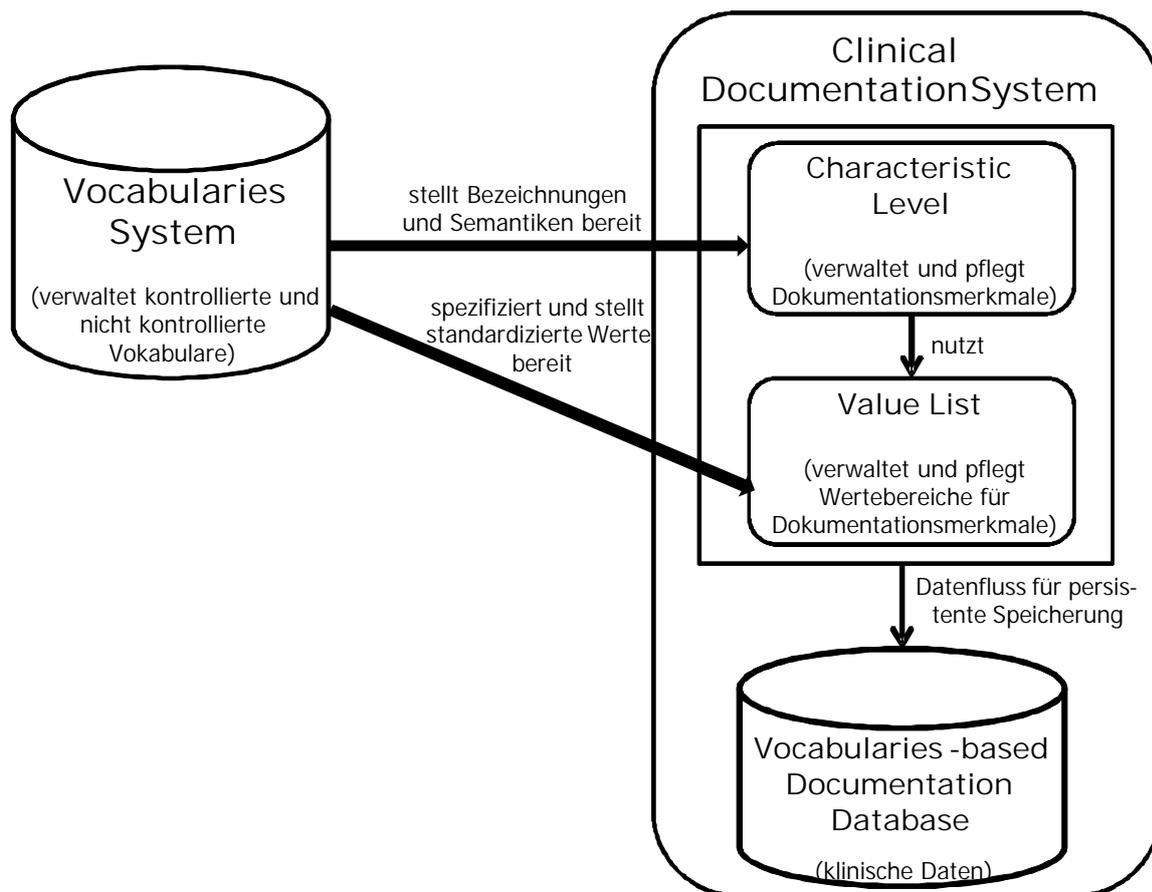
## **E.2 SPEZIFIKATION VON VOKADOKS**

### ***E.2.1 Einführung***

Zur Beschreibung eines eigenen Modells zur terminologischen Fundierung von Dokumentationssystemen werden die Analyse Kriterien berücksichtigt, die zur Bewertung ausgewählter bestehender Ansätze zur terminologischen Fundierung von Dokumentationssystemen festgelegt wurden. Das Modell wurde VokaDoks – Vokabular-basiertes medizinisches Dokumentationssystem – genannt und sollte an die Modellierung eines Dokumentationssystems unter Berücksichtigung von kontrollierten Vokabularen erinnern. Elemente von VokaDoks werden in den nachfolgenden Abschnitten kursiv geschrieben. Da einige Komponenten von VokaDoks sich an spezifischen Strukturen der ausgewählten analysierten Ansätze anlehnen, die in englischer Sprache veröffentlicht wurden, werden die VokaDoks-Komponenten in Englisch bezeichnet, sowie manche Beschreibungen auf Englisch gehalten.

### E.2.2 Konzept und Design

Die definierten Aufgaben bzw. Anforderungen zur Entwicklung eines Terminologie-basierten Dokumentationssystems wurden in drei Hauptsichten bzw. Module eingeteilt (s. Abbildung 14): Kontrolliertes Vokabular, Dokumentation und Datenbank. Diese drei Module interagieren folgendermaßen miteinander: Das klinische Dokumentationssystem (*Clinical Documentation System*) besteht aus Anwendungsebene (frontend) und Datenbankebene (backend). Auf Anwendungsebene wird der Merkmalskatalog implementiert bzw. werden Metadaten für die Erfassung medizinischer Dokumentationsmerkmale (*Characteristic Level*) auf Formularen dargestellt. Der *Characteristic Level* ist mit einer *Value List* verknüpft, um den Dokumentationsmerkmalen passende Merkmalsausprägungen zuordnen zu können. Auf Datenbankebene (*Vocabularies-based Documentation Database*) laufen Prozesse bzw. Prozeduren zur persistenten Speicherung der erfassten medizinischen Dokumentationsmerkmale. Das *Vocabularies System* dient dem *Clinical Documentation System* als Wissensbasis zur Standardisierung des Dokumentationssystems, indem es standardisierte Bezeichnungen und Werte, sowie Semantiken für die Dokumentationsmerkmale und deren Ausprägungen zur Verfügung stellt.



**Abbildung 14: Architektur des medizinischen Vokabular-basierten Dokumentationssystems (VokaDoks) mit Interaktionen zwischen seinen verschiedenen Modulen**

Aufbauend auf gewonnenes Wissen und abgeleitet von den Interaktionen zwischen den Modulen in Abbildung 14 wurde ein klinisches Informationsmodell unter Verwendung von medizinischen Vokabularen entworfen. Der Versuch beim Design dieses Informationsmodells war es, alle Stärken der fünf analysierten Ansätze zu berücksichtigen und im Informationsmodell umzusetzen (s. Abbildung 15).

Die für die vorliegende Arbeit relevante Stärke des ISO 11179 Ed. 3 Metamodells ist die Definition und Beschreibung von Dokumentationsmerkmalen sowohl in einer semantischen als auch in einer Repräsentationsebene, sowie die Bereitstellung von Modellen zur Einbindung von kontrollierten Vokabularen. So wurde beim Design des klinischen Informationssystems auf das *Data Description Package* and das *Concepts Package* des ISO 11179 Ed. 3 Metamodells zurückgegriffen, um die Merkmalsebene und die standardisierte Vokabularebene von VokaDoks zu modellieren.

In seinem Terminologie-Managementsystem verwaltet TERMTrial zwei Arten von Terminologien: allgemeine – darunter standardisierte – und studienspezifische Begriffe. Dies wird im VokaDoks-Informationsmodell durch die Komponenten *Controlled Vocabularies* und *Clinical Thesaurus* des *Vocabularies Systems* gespiegelt (s. Abbildung 15).

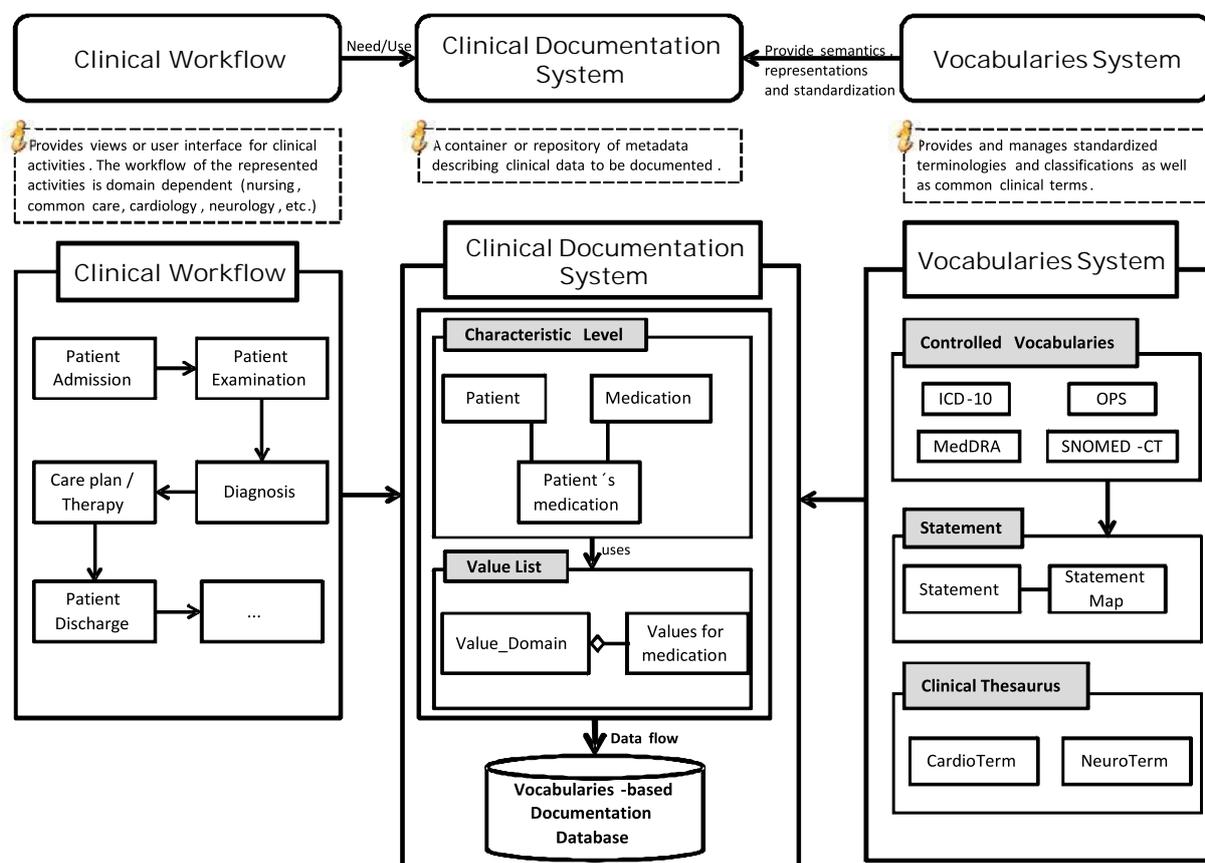
Was das Seoulsche Terminologie-basierte ENRS betrifft, stellen die „precoordinated statements“ ein gut auswertbares Konzept zur Erarbeitung neuer (standardisierter) medizinischer Begriffe dar, um den Anforderungen der medizinischen Dokumentation nachzukommen, z.B. mit der Implementierung eines möglichst umfangreichen Merkmalskatalogs.

Die Stärken von ORCA und InCoMe liegen insbesondere in der Beschreibung bzw. Spezifikation von Dokumentationsmerkmalen durch den Aufbau einer Wissensbasis mit einem medizinischen Thesaurus bzw. durch eine sorgfältige Beschreibung von Wertebereichen und Stammdaten für Dokumentationsmerkmale. Dieser Gesichtspunkt wird in VokaDoks durch das *Clinical Thesaurus* dargestellt. Speziell bei InCoMe stellt die Berücksichtigung von Integritätsbedingungen einen besonderen Beitrag dar, der in der vorliegenden Arbeit ebenfalls verwertet wurde.

### ***E.2.3 Das VokaDoks klinische Informationsmodell***

Abbildung 15 zeigt den konzeptuellen Datenfluss zwischen Front-end (*Clinical Documentation System* dem *Clinical Workflow* entsprechend) und Back-end (*Vocabularies-based Documentation Database und Vocabularies System*) in VokaDoks. Das *Clinical Documentation System* (klinisches Dokumentationssystem) zeigt den Inhalt eines klinischen Workflows zur Erfassung von medizinischen Daten. Die Dokumentationsaktivitäten im klinischen Workflow (*Clinical Workflow*) sind von der klinischen Domain und dem Dokumentationsprozess abhängig und können dementsprechend unterschiedlich sein. Die dargestellten Aktivitäten dienen daher nur als Beispiel. Der klinische Prozess umfasst unterschiedliche Arten von medizinischer Dokumentation in verschiedenen strukturierten und nicht strukturierten Formularen. Das *Clinical Documentation System* verwaltet ein Repository von Metadaten, die die durch den klinischen Prozess erfassten medizinischen Daten beschreibt. Das

*Clinical Workflow* benötigt bzw. setzt das *Clinical Documentation System* ein, um Spezifikationen von Dokumentationsmerkmalen abzufragen. Das *Vocabularies System* stellt für das *Clinical Documentation System* Semantiken und Darstellungsmöglichkeiten für Dokumentationsmerkmale bereit. Somit werden die Standardisierung und der terminologische Support für das *Clinical Documentation System* sichergestellt. Das *Vocabularies System* verwaltet sowohl standardisierte kontrollierte Vokabulare (standardisierte Terminologien und Klassifikationen) als auch medizinische präkoordinierte Datensätze (*Statement*) und domainspezifische Thesauri (*Clinical Thesaurus*), um die Aufnahme von standardisierten und nicht standardisierten bzw. spezifischen klinischen Begriffen zu ermöglichen. Nicht standardisierte und spezifische klinische Begriffe werden *Common Clinical Terms* genannt. Sämtliche medizinische Daten und Vokabulare werden in einer Vokabular-basierten Datenbank (*Vocabularies-based Documentation Database*) persistent gespeichert.



**Abbildung 15: Klinisches Informationsmodell für das Vokabular-basierte klinische Dokumentationssystem VokaDoks**

#### *E.2.4 Nutzung eines kontrollierten Vokabulars (terminologische Fundierung)*

Die terminologische Fundierung von VokaDoks erfolgt durch das *Vocabularies System* mit seinen drei Modulen *Controlled Vocabularies*, *Statement* und *Clinical Thesaurus*. Das *Vocabularies System* ermöglicht die Einbettung externer Standards wie medizinische Klassifikationen und Terminologien,

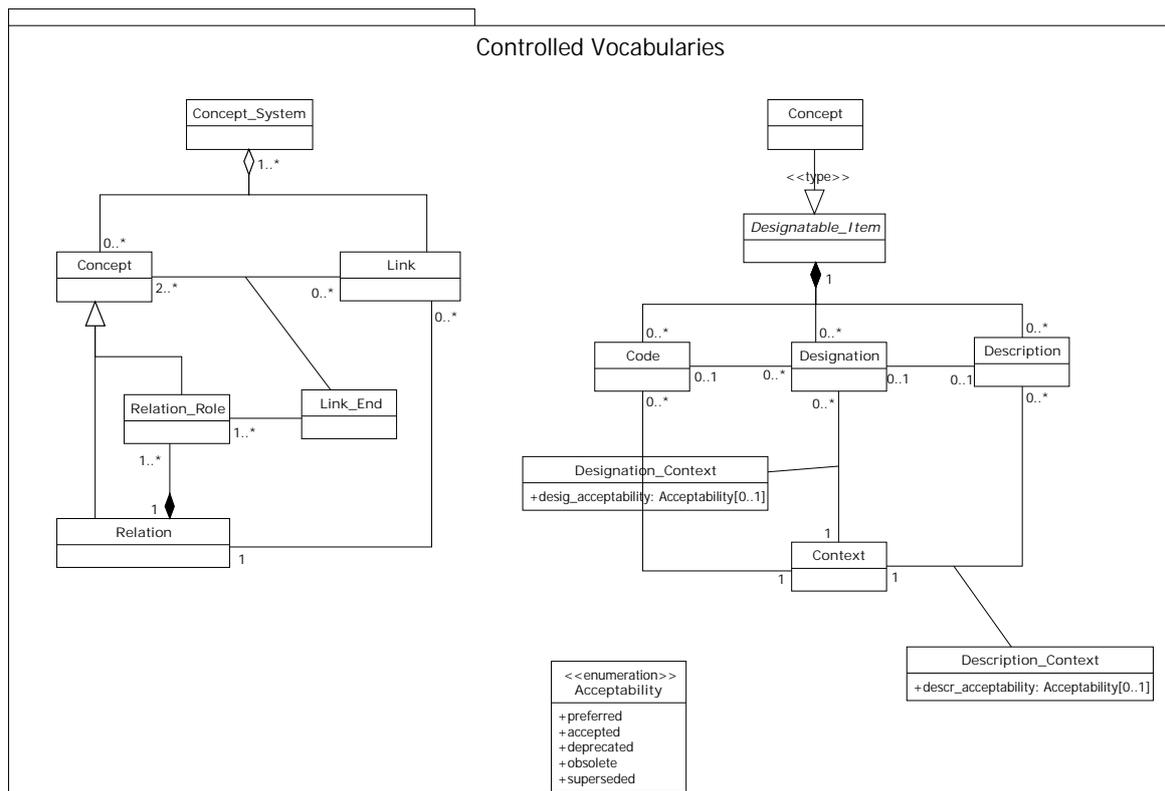
sowie die Integration von klinisch spezifischen Begriffen zur Definition, Beschreibung und Spezifikation von medizinischen Dokumentationsmerkmalen und ihren Wertebereichen.

#### E.2.4.1 Controlled Vocabularies

Das Modul *Controlled Vocabularies* des *Vocabularies Systems* dient der Verwaltung von standardisierten medizinischen kontrollierten Vokabularen, wie ICD-10, OPS, TNM, SNOMED-CT, etc. Der Aufbau dieses Moduls lehnt sich an die *Concept System Metamodel Region* des *Concepts Packages* des ISO 11179 Ed.3 Metamodells an (s. Abbildung 16). Dank dieser Struktur werden standardisierte medizinische kontrollierte Vokabulare im *Vocabularies System* von VokaDoks modelliert. Zum Beispiel werden für die Integration von ICD-10-GM 2013 zuerst die Informationen über diese Klassifikation über das ISO-Element *Concept\_System* registriert. Dieses *Concept\_System* besteht aus ICD-10-GM Klassen (die *Concepts*). Die hierarchische Strukturierung der Klassifikation wird über die ISO-Elemente *Relation*, *Link*, *Link\_End* und *Relation\_Role* abgebildet. Um sicherzustellen, dass alle Beziehungen zwischen Konzepten auch spezifisch zu dem *Concept\_System* sind, gehören alle *Links* zum *Concept\_System*. Jedes *Concept* ist vom ISO-Typ *Designatable\_Item* und kann als solches einen Kode (*Code*), eine Bezeichnung (*Designation*) und eine Beschreibung (*Description*) in einem spezifischen Kontext (*Context*) haben. Im *Context* wird die Nutzungsart (*Acceptability*) spezifiziert, also ob es sich bei einer *Designation* oder *Description* um eine „preferred, accepted, deprecated, obsolete oder superseded“ (bevorzugt, angenommen, überholt, veraltet oder ersetzt) Bezeichnung oder Beschreibung handelt.

#### Beispiel:

Das *Concept* „Cholera“ aus dem *Concept\_System* „ICD-10-GM 2013“ steht in einer monohierarchischen Beziehung zu dem *Concept* „Infektiöse Darmkrankheiten“, wobei die Klasse „Cholera“ eine Spezialisierung der Klasse „Infektiöse Darmkrankheiten“ ist. Als *Designatable\_Item* besitzt das *Concept* „Cholera“ die ‚preferred‘ *Designation* „Cholera“, die *Description* „schwere bakterielle Infektionskrankheit vorwiegend des Dünndarms, die durch das Bakterium *Vibrio cholerae* verursacht wird“ und den *Code* „A00.-“ im *Context* „Patientendiagnosen der inneren Medizin“.



**Abbildung 16: UML Klassendiagramm des Moduls *Controlled Vocabularies*, entsprechend dem Concept System Metamodel der ISO 11179 Ed. 3, zur Integration und Verwaltung von Standard medizinischen kontrollierten Vokabularen in VokaDoks**

#### E.2.4.2 Statement

Das Modul *Statement* des *Vocabularies Systems* dient der Erarbeitung und Verwaltung von zusammengesetzten bzw. präkoordinierten medizinischen Datensätzen aus medizinischen Begriffen aus vorhandenen kontrollierten Vokabularen (s. Abbildung 17). Die erzeugten *Statements* gehören zu den *Common Clinical Terms*. Die Zusammenführung zweier *Concepts* zur Erzeugung eines *Statements* erfolgt im Element *Statement\_Map*. In der *Statement\_Map* werden mindestens zwei *Concepts* aus gleichem oder unterschiedlichen *Concept\_Systems* präkoordiniert. Ein *Concept* kann mehrmals für eine Zusammenführung in der *Statement\_Map* verwendet werden und ein erzeugtes *Statement* hat eine oder mehrere *Statement\_Maps*. Ein *Statement* ist eine Spezialisierung eines *Concept*, das zu einem *Statement\_System* gehört. Das *Statement\_System* ist eine Spezialisierung eines *Concept\_System*. Jedes erzeugte *Statement* wird also in ein zuvor generiertes *Statement\_System* eingebunden.

Alternativ kann die Bildung von *Statements* durch die Nutzung von *Links* aus dem *Controlled Vocabularies* des *Vocabularies Systems* erfolgen.

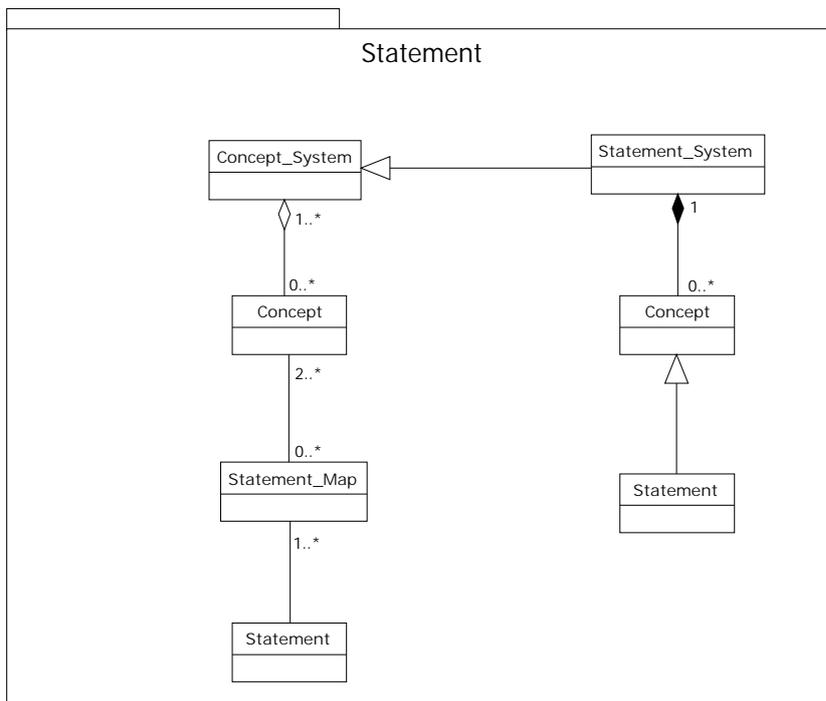
#### Beispiel 1:

Das *Statement* „Klinische Untersuchung in Allgemeinanästhesie bei indeterminierter Lepra“ setzt sich aus dem *Concept* „Klinische Untersuchung in Allgemeinanästhesie“ (*Code*: A30.0) aus dem

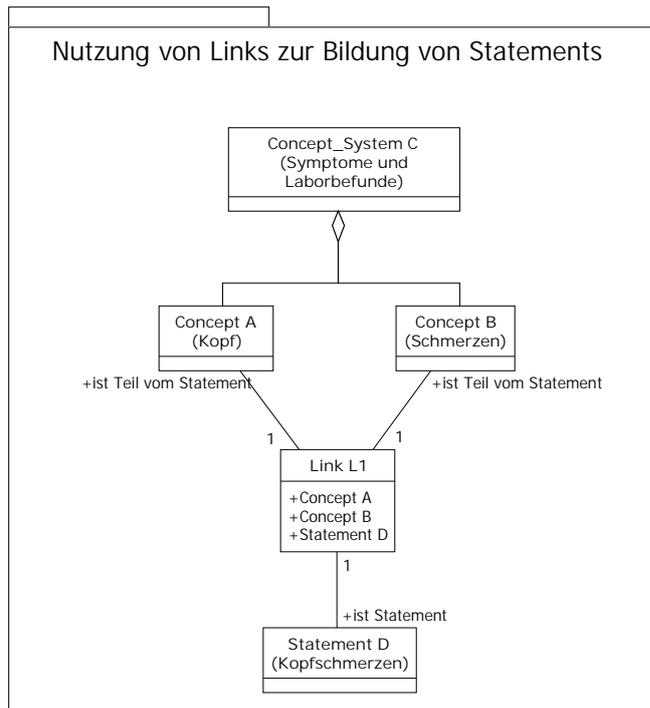
*Concept\_System* „ICD-10-GM 2013“ und dem *Concept* „Indeterminierte Lepra“ (*Code*: 1-100) aus dem *Concept\_System* „OPS 2013“ zusammen. Das erzeugte *Statement* wird zum Beispiel als *Concept* in das *Statement\_System* „Diagnostische Maßnahmen bei bakteriellen Krankheiten“ eingebunden.

Beispiel 2:

Seien *Concept A* „Kopf“ und *Concept B* „Schmerzen“ zwei Konzepte aus dem *Concept\_System C* „Symptome und Laborbefunde“ und das *Statement D* „Kopfschmerzen“. Zur Bildung von *Statement D* werden *Concept A*, *Concept B* und *Statement D* miteinander durch den *Link L1* verlinkt, wobei *Concept A* und *Concept B* jeweils die *Relation\_Role* „ist\_Teil\_vom\_Statement“ haben und *Statement D* die *Relation\_Role* „ist\_Statement“ (s. Abbildung 18).



**Abbildung 17: UML Klassendiagramm des Moduls *Statement* zur Erarbeitung und Verwaltung von präkoordinierten medizinischen Begriffen in VokaDoks**



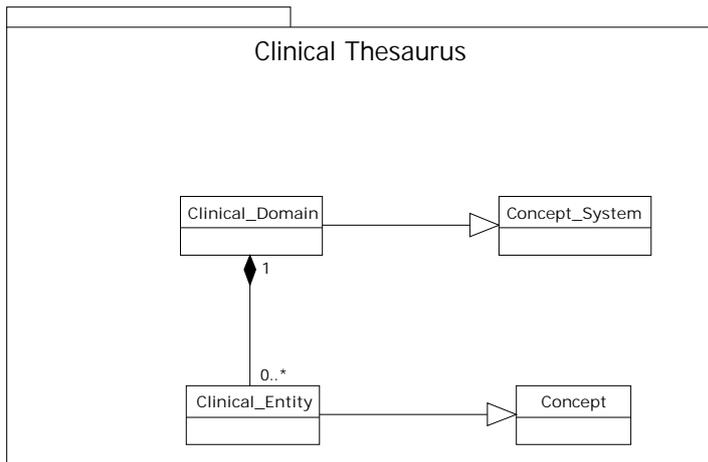
**Abbildung 18: UML Klassendiagramm zur Darstellung der Nutzung von *Links* zur Bildung von *Statements* als Alternative zum *Statement\_Map***

#### E.2.4.3 Clinical Thesaurus

Das Modul *Clinical Thesaurus* des *Vocabularies Systems* dient der Erarbeitung und Verwaltung von medizinischen Begriffen, die in einem klinischen Bereich (z.B. Urologie) sehr spezifisch sind und weder in medizinischen kontrollierten Vokabularen zu finden sind noch sich aus gleichen oder unterschiedlichen vorhandenen Vokabularen zusammensetzen lassen. Jeder klinisch spezifische Begriff ist ein *Clinical\_Entity*. Ein *Clinical\_Entity* ist ein *Concept* und gehört zu einer bestimmten *Clinical\_Domain*. Eine *Clinical\_Domain* ist eine Spezialisierung eines *Concept\_Systems* und besteht aus mehreren *Clinical\_Entities* (s. Abbildung 19).

#### Beispiel:

Im Bereich der Urologie wird zum Beispiel der (nicht standardisierte) Begriff „Blasenkatheter, transurethral“ häufig verwendet. Dieser Begriff wird als *Clinical\_Entity* erzeugt und gehört zu der *Clinical\_Domain* „UroTerm“.



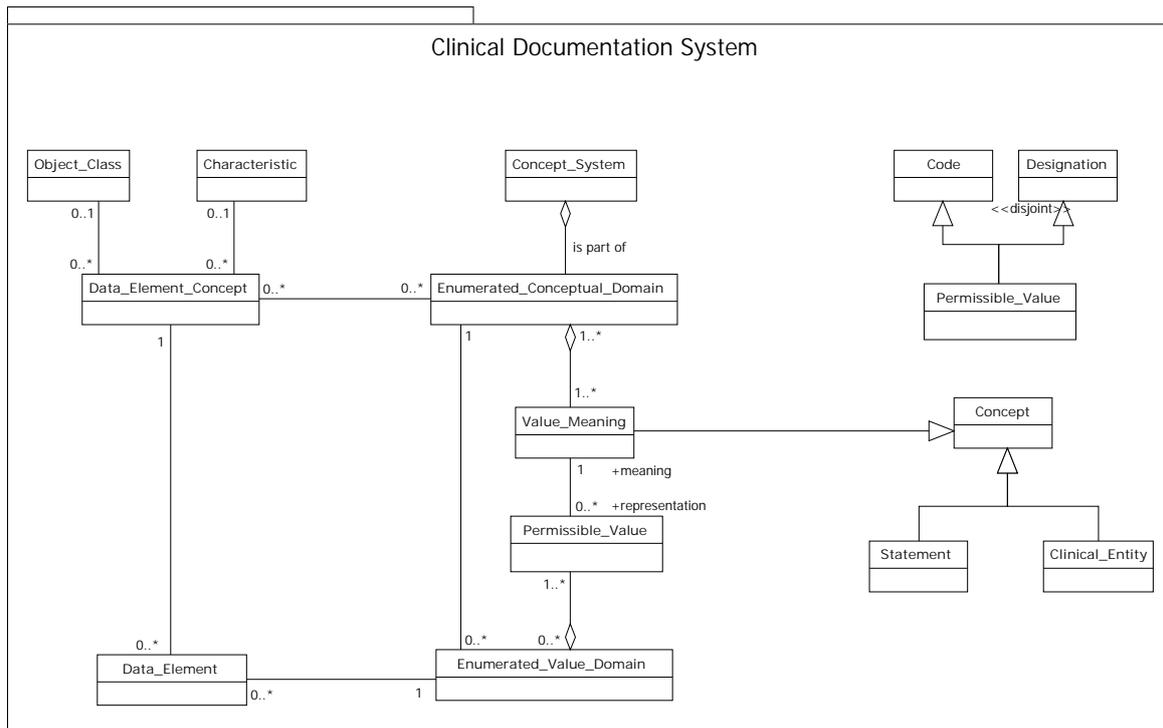
**Abbildung 19: UML Klassendiagramm des Moduls *Clinical Thesaurus* zur Erarbeitung und Verwaltung von spezifischen bzw. domainabhängigen klinischen Begriffen in VokaDoks**

### ***E.2.5 Merkmalsebene***

Die Modellierung der Merkmalsebene in VokaDoks lehnt sich an das Data Description Metamodel der ISO 11179 Ed. 3 an. Das Modul *Characteristic Level* des *Clinical Documentation Systems* dient der Modellierung von Dokumentationsmerkmalen (s. Abbildung 20).

Ein Dokumentationsmerkmal wird durch das *Data\_Element* dargestellt. In der semantischen Ebene wird dieses *Data\_Element* durch ein und nur ein bestimmtes Konzept, das *Data\_Element\_Concept*, dargestellt. Ein *Data\_Element\_Concept* kann wiederum in der Darstellungsebene in der Form eines *Data\_Elements* dargestellt werden. Ein *Data\_Element\_Concept* kann mehreren *Data\_Elements* zugeordnet werden. Die Verbindung *Object\_Class* und *Characteristic* ergibt ein *Data\_Element\_Concept*. Eine *Object\_Class* ist nichts anderes als eine Dokumentationseinheit bzw. ein Merkmalsträger – das Objekt von Interesse – deren Merkmale beschrieben bzw. dokumentiert werden sollen. *Characteristic* ist die Merkmalsart der Dokumentationseinheit.

Das *Data\_Element\_Concept* verleiht einem *Data\_Element* eine (standardisierte) Bedeutung in einem bestimmten semantischen Bereich (*Enumerated\_Conceptual\_Domain*) und bietet der *Enumerated\_Conceptual\_Domain* wiederum einen Anwendungskontext. Zum Beispiel könnte die *Enumerated\_Conceptual\_Domain* „Länder der Welt“ in folgenden Anwendungskontexten bzw. *Data\_Element\_Concepts* benutzt werden: „Aufenthaltsland einer Person“, „Geburtsland einer Person“, oder „Staatsangehörigkeit einer Person“. Die *Enumerated\_Conceptual\_Domain* definiert ein Set von aufgezählten Semantiken für Werte (*Value\_Meaning*) bzw. Merkmalsausprägungen.



**Abbildung 20: UML Klassendiagramm zur Darstellung der Merkmalsebene und Wertebereiche für die Verwaltung von Dokumentationsmerkmalen in VokaDoks**

### E.2.6 Wertebereiche

Die Modellierung von Wertebereichen in VokaDoks lehnt sich an das Data Description Metamodel der ISO 11179 Ed. 3. Das Modul *Value List* des *Clinical Documentation Systems* dient der Modellierung von Wertebereichen für Dokumentationsmerkmale (s. Abbildung 20).

Jedem *Data\_Element* aus der Merkmalsebene wird ein und nur ein Wertebereich (*Enumerated\_Value\_Domain*) zugeordnet. Die *Enumerated\_Value\_Domain* besteht aus zulässigen Werten bzw. Ausprägungen für Dokumentationsmerkmale, den sogenannten *Permissible\_Values*. *Permissible\_Values* können Bezeichnungen (*Designation*) oder Zeichen, Kodierungen (*Code*) zu ihren entsprechenden Semantiken (*Value\_Meaning*) sein. Während *Value\_Meanings* Semantiken bzw. Bedeutungen für Werte bereitstellen, bieten *Permissible\_Values* eine Darstellung bzw. Repräsentation für diese Semantiken. Zum Beispiel könnte eine mögliche/zulässige Darstellung für die *Value\_Meaning* „Deutschland“ aus der *Enumerated\_Conceptual\_Domain* „Länder der Welt“ „DE“ oder „de“ sein. Ebenso werden durch die Beziehung zwischen einer *Enumerated\_Value\_Domain* und einer bestimmten *Enumerated\_Conceptual\_Domain* jeweils mehrere Darstellungsmöglichkeiten für die *Enumerated\_Conceptual\_Domain* und ein bestimmter semantischer Bereich für die *Enumerated\_Value\_Domain* geboten. Die *Enumerated\_Conceptual\_Domain* kann Teil eines *Concept\_System* sein.

Eine *Value\_Meaning* kann als Spezialisierung eines *Concepts* ein *Statement* aus dem *Statement\_System*, ein *Clinical\_Entity* aus der *Clinical\_Domain* oder ein Konzept aus einem standardisierten *Concept\_System* sein. Somit wird nochmal die relevante Rolle von medizinischen kontrollierten Vokabularen in VokaDoks gewährleistet, in dem sie durch *Value\_Meaning* Semantiken für Werte und Anwendungskontexte bereitstellen und somit für Standardisierung sorgen. Falls ein *Permissible\_Value* ein *Code* oder eine *Designation* aus dem Modul *Controlled Vocabularies* des *Controlled Vocabularies Systems* ist, wird auch hier eine Standardisierung der *Enumerated\_Value\_Domain* gewährleistet.

### **E.2.7 Rollen und Entitätstypen**

Jedes Objekt des Moduls *Controlled Vocabularies* des *Vocabularies Systems* ist ein Element mit folgenden Rollen: *Concept\_System*, *Concept*, *Relation*, *Link*, *Link\_End*, *Relation\_Role*. Ein *Concept\_System* besteht aus mehreren *Concepts*, die miteinander mit *Links* verbunden werden und in jeder bestimmten *Relation* auch bestimmte *Relation\_Roles* haben. Als *Designatable\_Item* kann ein *Concept* einen *Code*, eine *Designation* und/oder eine *Description* in einem bestimmten *Context* haben. Jedes Objekt des Moduls *Statement* des *Vocabularies Systems* ist ein Element mit folgenden Rollen: *Statement\_System*, *Statement* und *Statement\_Map*. Ein *Statement\_System* besteht aus mehreren *Statements*, deren Bildung in der *Statement\_Map* erfolgt.

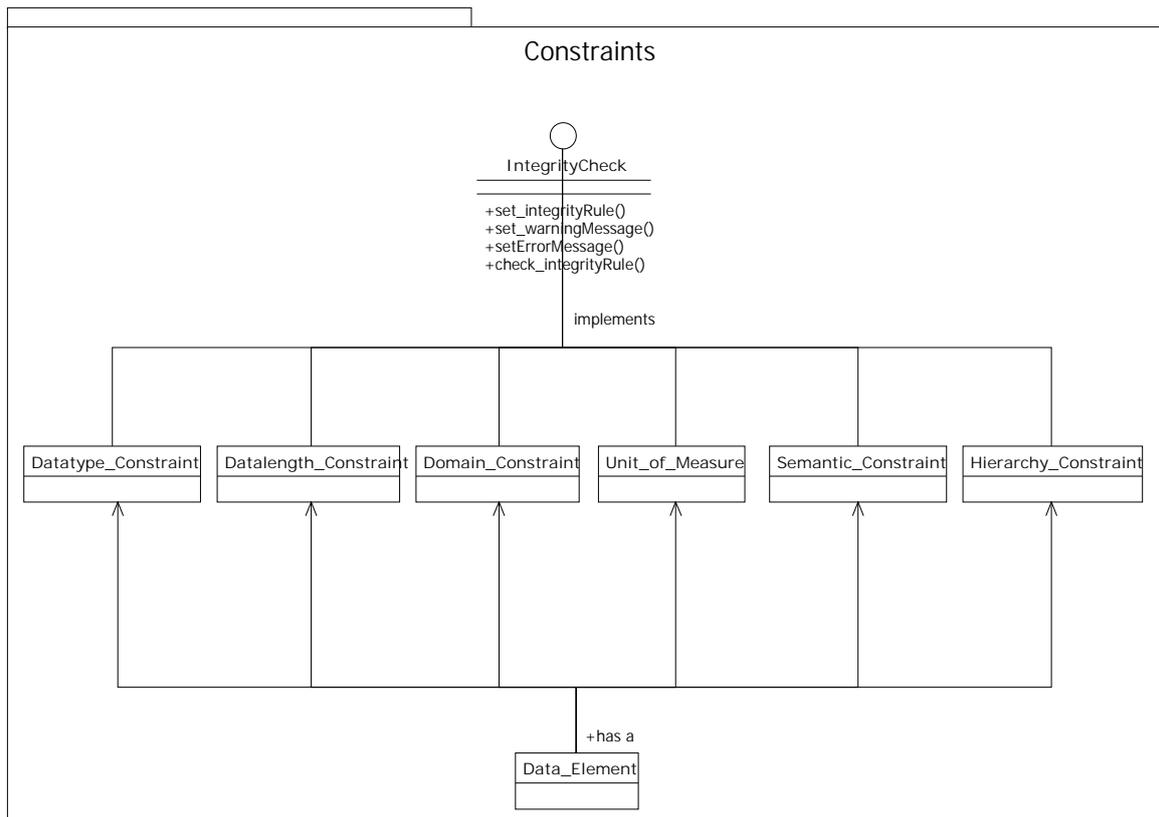
Jedes Objekt des Moduls *Clinical Thesaurus* des *Vocabularies Systems* ist ein Element mit folgenden Rollen: *Clinical\_Entity* und *Clinical\_Domain*. Eine *Clinical\_Domain* besteht aus mehreren *Clinical\_Entities*.

Jedes Objekt des Moduls *Characteristic Level* des *Clinical Documentation Systems* ist ein Element mit folgenden Rollen: *Object\_Class*, *Characteristic*, *Data\_Element\_Concept*, *Data\_Element*, *Enumerated\_Conceptual\_Domain*, *Value\_Meaning*. *Data\_Element* und *Value\_Domain* werden benutzt, um Dokumentationsmerkmale und ihre Ausprägungen (Wertebereiche) zu beschreiben.

Jedes Objekt des Moduls *Value List* des *Clinical Documentation Systems* ist ein Element mit folgenden Rollen: *Enumerated\_Value\_domain* und *Permissible\_Value*. Eine *Enumerated\_Value\_domain* besteht aus mehreren *Permissible\_Values*.

### **E.2.8 Constraints**

Der Schwerpunkt bei dem Dokumentationsmodell InCoMe liegt in der Formulierung von Integritätsbedingungen zur Sicherstellung der Datenqualität. Von den zehn im InCoMe formulierten Integritätsbedingungen wurden folgende fünf Integritätsbedingungen für *Data\_Element* bei der Modellierung von VokaDoks berücksichtigt: Datentyp-, Datenlänge-, Wertebereich-, Hierarchie- und semantische Integritätsbedingung. Eine weitere Integritätsbedingung wurde hinzugefügt: die *Unit\_of\_Measure*-Integritätsbedingung, die die Maßeinheit eines Dokumentationsmerkmals festlegt (s. Abbildung 21). Verletzungen dieser Integritätsbedingungen werden im Dokumentationssystem entsprechend behandelt.



**Abbildung 21: UML Klassendiagramm zur Darstellung von unterschiedlichen Typen von Integritätsbedingungen in VokaDoks**

### ***E.2.9 Abschnittszusammenfassung***

In der vorliegenden Arbeit wurde das Dokumentationsmodell VokaDoks vorgestellt, das die Entwicklung eines terminologisch fundierten Dokumentationssystems erleichtern soll. Dafür liefert VokaDoks drei wichtige Strukturen: Das *Vocabularies System* zur Verwaltung von Standard Terminologien und Klassifikationen, sowie von nicht standardisierten medizinischen Begriffen, das *Clinical Documentation System* zur Modellierung von Dokumentationsmerkmalen und Wertebereichen für Dokumentationsmerkmale, und die *Vocabularies-based Documentation Database* zur persistenten Speicherung der erzeugten Dokumentationsmerkmale mit ihren Bezeichnungen und Ausprägungen aus Konzepten des *Vocabularies Systems*. Diese Hauptkomponenten von VokaDoks erfüllen die im Abschnitt C.4 definierten Analyse- und Qualitätskriterien eines terminologisch fundierten medizinischen Dokumentationssystems, nämlich und vor allem die Nutzung eines kontrollierten Vokabulars zur Definition und/oder Beschreibung von Merkmalen, der Aufbau einer Merkmalsebene zur Erzeugung und Pflege von Merkmalen, die Definition von Wertebereichen bzw. Wertelisten für die Merkmale, die Definition von Rollen und Enttytätstypen je nach Anwendungsgebiet bzw. je nach medizinischem Kontext der Merkmale und zusätzlich die Definition von

Integritätsbedingungen sogenannten Constraints zur Überprüfung der Konsistenz und Exaktheit der Daten (s. Tabelle 3).

Nun soll zur Rechtfertigung der Umsetzbarkeit von VokaDoks eine prototypische Implementierung des Dokumentationsmodells erfolgen.

**Tabelle 3: Erfüllungsstatus der Prüfkriterien für eine terminologische Fundierung**

	<b>Vokabular</b>	<b>Rollen</b>	<b>Merkmalswelt</b>	<b>Speicher für Wertebereiche</b>	<b>Integritätsbedingungen Für Merkmale</b>
VokaDoks- Modell	✓	✓	✓	✓	✓

## F PROTOTYPISCHE IMPLEMENTIERUNG

---

### F.1 ARCHITEKTUR UND PROGRAMMIERUNG

#### F.1.1 Allgemeines

Zur prototypischen Implementierung wird das InCoMe- Entwurfsmodul zu Grunde gelegt und ggf. mit Umbenennungen, sowie neuen Bezeichnungen und Modellelementen und vor allem mit kontrollierten Vokabularen erweitert, um die Anforderungen des VokaDoks zu erfüllen. Das InCoMe- Entwurfsmodul, auch Designer genannt, unterstützt den Entwurf eines Dokumentationssystems über eine grafische Benutzeroberfläche. Der terminologische Ausbau dieses Designers sollte zur Modellierung eines VokaDoks-Dokumentationssystems dienen und wird daher in VokaDoks-Designer umbenannt.

Vor diesem Ausbau ist jedoch eine Abbildung von InCoMe-Modellelementen im VokaDoks-Dokumentationsmodell erforderlich. Dadurch werden die notwendigen Ausbaupunkte deutlich sichtbar.

#### F.1.2 Grundlegende Aspekte

Trotz der Tatsache, dass InCoMe die Einbindung von kontrollierten Vokabularen bei der Modellierung von Dokumentationssystemen im CDMS-Designer nicht vorsieht, wurde entschieden, auf sein Dokumentationsmodell aufzubauen. Zum einen wurde diese Entscheidung getroffen, weil InCoMe durch seinen CDMS-Designer ein attraktives Werkzeug für die Modellierung von Plattform- und Datenbankmanagementsystem-unabhängigen Dokumentationssystemen bietet, zum anderen, weil sämtliche Source-codes von InCoMe für das Gelingen dieser Arbeit zur Verfügung gestellt wurden. VokaDoks basiert außerdem auf dem ISO/IEC 11179 Ed.3 Standard, der zwei mächtige Datenmodelle zur Modellierung von Dokumentationsmerkmalen (die genannten Datenelemente) und zur Modellierung von Konzepten aus kontrollierten Vokabularen zur Verfügung stellt: das *Data Description Metamodel* und das *Concept System Metamodel*.

Während InCoMe also eine einfache Modellierung von Dokumentationsmerkmalen bietet, bietet das *Concept System Metamodel* des ISO/IEC 11179 Ed.3 Standards eine ausdrucksvolle Modellierung von Vokabularen für die Standardisierung von Dokumentationsmerkmalen. Zudem würde das *Data Description Metamodel* das Dokumentationsmodell von InCoMe ausdrucksvoll, effizient und harmonisch ergänzen und standardisieren.

Ein Mapping von Elementen des Dokumentationsmodells von InCoMe zu Modellelementen des *Data Description Metamodels* des ISO/IEC 11179 Ed.3 Standards, sowie Überlegungen zur Verbindung von InCoMe-Modellelementen zur Beschreibung von Dokumentationsmerkmalen mit Modellelementen des *Concept System Metamodels* des ISO/IEC 11179 Ed.3 Standards so, wie sie im VokaDoks-Dokumentationsmodell dargestellt wurden, würde daher die prototypische Implementierung eines VokaDoks-basierten Dokumentationssystems erleichtern.

### F.1.3 Abbildung von InCoMe in ISO/IEC 11179 Ed.3

Zwischen dem InCoMe-Dokumentationsmodell und dem ISO/IEC 11179 Ed. 3 Metamodell bestehen Ähnlichkeiten, vor allem wenn man das *Data Description Metamodel* und das *Concept System Metamodel* des ISO/IEC 11179 Ed.3 Standards betrachtet. Allerdings bleibt ISO/IEC 11179 Ed.3 mächtiger, da der Standard alle Modellelemente von InCoMe einkapselt. Die Art der Gruppierung und die klar definierten Integritätsbedingungen von InCoMe scheinen aber vorteilhafter.

Im Folgenden soll die Abbildung von Elementen des InCoMe-Dokumentationsmodells im *Data Description Metamodel* des ISO/IEC 11179 Ed.3 Standards beschrieben werden. Tabelle 4 fasst dabei die Ergebnisse dieses Mappings zusammen. Um die Komplexität beider Systeme etwas zu erleichtern, erfolgt der Vergleich elementenweise. Bei diesem Vergleich werden ISO-Modellelemente zwischen << >> und InCoMe-Elemente kursiv geschrieben.

#### F.1.3.1 Entity (InCoMe) – Designatable\_Item (ISO), Object (ISO)

Das Element *Entity* ist eine modulare Einheit bzw. ein abstraktes Element, das gemeinsame Eigenschaften aller InCoMe-Modellelemente zusammenfasst. Jedes Element in ISO kann ein <<Designatable\_Item>> sein. Da die Eigenschaften einer *Entity* u.a. aus Bezeichnung, Name und Beschreibung bestehen, kann man dies leicht mit <<Designatable\_Item>> abbilden, das <<Designation>> und/oder <<Definition>> haben kann.

#### F.1.3.2 Freie Merkmalsart (InCoMe) – Characteristic (ISO)

*Freie Merkmalsarten* sind in InCoMe Instanzen vom Typ *Attribute*, d.h. definieren Charakteristiken von Objekten (Merkmalsträgern). Somit passt die Klasse <<Characteristic>> von ISO/IEC 11179 Ed.3 hier sehr gut. Außerdem ist <<Characteristic>> in ISO/IEC 11179 Ed.3 Instanz von <<Concept>>. Folglich können *Attribute* auch <<Concept>> sein.

#### F.1.3.3 MasterData/Stammdaten (InCoMe) – Enumerated\_Value\_Domain (ISO)

*Stammdaten* bzw. *MasterData* sind Wertelisten bzw. Ausprägungslisten von freien Merkmalsarten. Jeder Wert in *Stammdaten* stellt ein <<Permissible\_Value>> in ISO/IEC 11179 Ed.3 dar. In InCoMe gibt es allerdings keine entsprechende Klasse für <<Permissible\_Value>>, da Werte von Stammdaten manuell eingegeben werden, um Stammdaten unmittelbar zu erstellen. Als Set von möglichen bzw. zulässigen Werten entsprechen daher Stammdaten der <<Enumerated\_Value\_Domain>> der ISO/IEC 11179 Ed.3.

#### F.1.3.4 ContextAttribute/gebundene Merkmalsarten (InCoMe) – Data\_Element (ISO)

Während in InCoMe die zentrale Bedeutung für die Modellierung eines Dokumentationssystems die Menge der darin zu dokumentierenden Informationen hat, spricht die Merkmalsarten, besitzt das <<Data\_Element>> die zentrale Bedeutung, indem Mittel zur Verfügung gestellt werden, um <<Data\_Element>> möglichst genau zu beschreiben.

#### F.1.3.5 Context (InCoMe) – Data\_Element\_Concept (ISO)

Ein *Context* ist in InCoMe die Menge von semantisch zusammengehörenden Merkmalsarten. Durch den *Context* werden diese Merkmalsarten zusammengebunden (daher *gebundene Merkmalsarten* bzw. *ContextAttribute* genannt). Beispiel: zu dem Context „Patient“ können folgende Merkmalsarten gehören: Geschlecht, Geburtsdatum, Aufnahme datum, etc.). Diese Funktion (Merkmalsträger) übernimmt in ISO/IEC 11179 Ed.3 die Klasse <<Data\_Element\_Concept>> zu der mehrere <<Data\_Element>> gehören.

#### F.1.3.6 Database (InCoMe) – Context (ISO), Conceptual\_Domain (ISO), Concept\_System (ISO)

Eine *Database* (Datenbank) in InCoMe ist die Menge aller *Contexts* bzw. aller Dokumentationsobjekte eines medizinischen Dokumentationssystems. In einer *Database* können zudem alle *Contexts* hierarchisch gegliedert werden. Das Element *Database* entspricht einem <<Context>>, einer <<Conceptual\_Domain>> oder einem <<Concept\_System>> in ISO/IEC 11179 Ed.3, da Objekte aus diesen ISO-Elementen semantisch korrelieren und ggf. in Beziehung zueinander stehen.

#### F.1.3.7 Constraint (InCoMe) – Described\_Value\_Domain (ISO), Assertion (ISO), Relation (ISO)

*Constraint* stellen in InCoMe Einschränkungen bzw. logische Formeln dar, zur Sicherstellung der Integrität (Korrektheit und Vollständigkeit) der im medizinischen Dokumentationssystem erfassten Daten. Somit wird zum Beispiel verhindert, dass bei einem Merkmalsträger (*Context*) „Patient“ dessen Eigenschaft (*gebundene Merkmalsart*) „Schwangerschaft“ mit dem Wert „Ja“ belegt wird, wenn es sich bei dem Patient um einen Mann handelt (Eigenschaft „Geschlecht“ = „männlich“). Diese sind sogenannte *semantische Constraints* entsprechend <<Assertion>> und <<Relation>> in ISO/IEC 11179 Ed.3.

In InCoMe gibt es allerdings auch *Constraints*, die Wertebereiche quantitativer Merkmalsarten beschreiben (z.B. Länge, gültiger Bereich, Datentyp, etc.). Diese Art von *Constraints* entspricht der <<Described\_Value\_Domain>> in ISO/IEC 11179 Ed.3.

**Tabelle 4: Zusammenfassung des Mapping von InCoMe-Modellelementen zu ISO/IEC 11179 Ed.3 Modellelemente**

InCoMe Modellelemente	ISO/IEC 11179 Ed.3	Begründung/Kommentar
Entity	Designatable_Item, Objekt	Abstraktes Konstrukt von dem Dokumentationseinheiten abgeleitet werden können.
Freie Merkmalsart	Characteristic	Beschreibende Eigenschaft eines Objekts (Merkmalsträger)
MasterData (Stammdaten)	Enumerated_Value_Domain	Menge aller möglichen

InCoMe Modellelemente	ISO/IEC 11179 Ed.3	Begründung/Kommentar
		Werte/Ausprägungen einer Merkmalsart bzw. eines Data_Element
ContextAttribute (Gebundene Merkmalsart)	Data_Element	Dokumentationseinheit, deren Semantik durch den Context/Data_Element_Concept definiert wird.
Context (Merkmalsträger)	Data_Element_Concept	Semantischer Träger mehrerer Merkmalsarten/Data_Element
Database (Datenbank)	Context, Conceptual_Domain, Concept_System	Konzeptueller Container aller Objekte eines Dokumentationssystems
Constraint (Integritätsbedingung)	Described_Value_Domain, Assertion, Relation	Semantische Integritätsbedingungen, Beschreibung von Wertebereichen quantitativer Merkmalsarten/Data_Element
X	Permissible_Value	InCoMe definiert kein bestimmtes Konstrukt für einzelne Werte von Stammdaten.
X	Value_Meaning	Das Konzept von Value_Meaning existiert in InCoMe nicht.

#### **F.1.4 InCoMe-Ausbau**

Abbildung 22 fasst die Ergebnisse der Abbildung von Elementen des InCoMe-Dokumentationsmodells im *Data Description Metamodel* des ISO/IEC 11179 Ed.3 Standards zusammen.

Anhand dieser Ergebnisse werden folgende Erweiterungen im InCoMe-Designer implementiert:

##### **F.1.4.1 Bezeichnung von Merkmalsarten und Stammdaten**

Es soll dem Benutzer ermöglicht werden, standardisierte Namen für die Bezeichnungen von Dokumentationseinheiten zu übernehmen.

#### F.1.4.2 Freie Merkmalsart

Die Erfassung der freien Merkmalsarten wird im Designer um eine weitere Eigenschaft erweitert. Es soll die Erfassung von Messeinheiten von freien Merkmalsarten ermöglicht werden (Unit\_of\_Measure\_Constraint). Beispiel: Die freie Merkmalsart „Blutdruck“ hat einen Wertebereich (Wertebereich-Integritätsbedingung) von 100/60 bis 130/85 mit der Messeinheit mmHg, deren Bezeichnung Millimeter Quecksilbersäule lautet. Abbildung 23 zeigt die bisherige Ansicht zur Erstellung von freien Merkmalsarten.

#### F.1.4.3 Stammdaten (MasterData)

Die Erfassung der benötigten Stammdaten wird durch kontrollierte Vokabulare gesteuert.

In der bisherigen Version des InCoMe-Designers (s. Abbildung 24) werden Stammdaten ohne Nutzung bzw. Zugriff auf kontrollierte Vokabulare erstellt. Die Bezeichnung sowie einzelne Werte von Stammdaten werden manuell eingegeben. Diese Werte können dann beliebig sortiert werden. Die Erweiterung in der Erstellung von Stammdaten besteht darin, standardisierte Werte aus kontrollierten Vokabularen für die Stammdaten zur Verfügung zu stellen. Durch eine Suchfunktion soll dem Benutzer ermöglicht werden, einzelne Werte für seine benötigten Stammdaten in der Vokabular-Datenbank zu suchen und anschließend auszuwählen. Durch manuelle Eingabe eines bestimmten Begriffs hat der Benutzer ebenfalls die Möglichkeit, die Vokabular-Datenbank nach diesem Begriff zu durchsuchen. Sämtliche Konzepte, die diesen Begriff beinhalten, werden ihm dann zur Auswahl angeboten. Die Suche kann in der gesamten Vokabular-Datenbank oder in einem bzw. in vorab ausgewählten Vokabularen erfolgen (z.B. Suche vom Begriff „Kopfschmerzen“ lediglich in ICD-10-GM 2012). Da einzelne Werte von Stammdaten *Permissible\_Value* sind, und ein *Permissible\_Value* ein *Code* oder (XOR=exklusives oder) eine *Designation* sein kann, kann der Benutzer zum Schluss auswählen, ob er den Code oder die Bezeichnung des gesuchten Konzepts in seinen zu erstellenden Stammdaten übernehmen möchte.

Zum Nachverfolgen oder Nachvollziehen des Ursprungs einzelner Werte von Stammdaten werden UUIDs benutzt. Jeder Wert führt seine UUID mit.

### ***F.1.5 Technische Anforderungen und Entwicklungsumgebung***

#### F.1.5.1 Einführung

Der InCoMe-Ausbau legt die in InCoMe angewandten Entwicklungs-Technologien zu Grunde, benutzt jedoch ihre aktualisierten Versionen. Die Weiterimplementierung des InCoMe-Designers, in VokaDoks-Designer umbenannt, basiert auf Java und das resultierende Dokumentationssystem auf J2EE-Webentwicklung. Die einwandfreie Ausnutzung des zu entwickelnden VokaDoks-Designers sowie die Ausführung der Dokumentationssystem-Webanwendung erfordern bestimmte Systemeinstellungen (sogenannte prerequisites) sowie die Installation bestimmter Softwarepakete. Im Folgenden werden sie genauer erläutert.

#### F.1.5.2 Java-Einstellungen

Als Programmiersprache für die Implementierung dient Java in seiner Version SDK 1.7. Als integrierte Entwicklungsumgebung (IDE) für Java wird das Programmierwerkzeug Eclipse-Juno benutzt. Der VokaDoks-Designer benötigt als Java-basierte Benutzeroberfläche das Vorhandensein einer Java Laufzeitumgebung (Java Runtime Environment, JRE) auf dem Betriebssystem. Die Version `jdk1.7.0_51` wurde installiert. Das Software Development Kit (SDK) für die Implementierung und Ausführung von Java Anwendungen lässt sich schnell unter <http://www.oracle.com/technetwork/java/javase/downloads/index.html> herunterladen. Damit Java vom Betriebssystem und von anderen laufenden Anwendungen erkannt wird, sollen Umgebungsvariablen gesetzt werden. So stellt man sie zum Beispiel ein: `JAVA_HOME = C:\Programme\Java\jdk1.7.0_51` und `path= C:\Programme\Java\jdk1.7.0_51\bin`.

Zur Generierung von XML-Konfigurationsdateien aus dem VokaDoks-Designer, welche für die Installation des Dokumentationssystems erforderlich sind, werden spezifische Java Archive (.jar) benötigt. Diese Archive sollen in das Verzeichnis kopiert werden, in dem JRE installiert wurde, zum Beispiel in `C:\Programme\Java\jre7\lib\ext\`.

Das Java Document Object Model (JDOM) ist eine weitere Java-Bibliothek, welche ermöglicht, XML Daten über Java-Objekte einzulesen, zu manipulieren und wieder zu schreiben. JDOM bietet auch die Möglichkeit strukturierte Daten als XML-Dokument auszugeben. Dazu benötigt die JDOM-Bibliothek den XSLT-Prozessor Xalan und den XML-Parser Xerces. Unter [www.jdom.org](http://www.jdom.org) lässt sich das ganze jdom Paket herunterladen sowie die online Dokumentation lesen. Wichtig dabei ist, das `jdom.jar` Archiv zu extrahieren, in das Verzeichnis `\lib\ext\` des Installationsverzeichnisses des JRE zu kopieren und dem Klassenpfad des Systems (Umgebungsvariablen) hinzuzufügen.

Nach der Modellierung mit dem VokaDoks-Designer und vor Ausführung der CDMS-Anwendung muss jedoch das `jdom`-Archiv ins Webcontainer-Installationsverzeichnis kopiert werden (am Beispiel Tomcat unter `C:\Program Files\Tomcat 7.0\lib`).

#### F.1.5.3 xalan.jar und xerces.jar

Der XSLT-Prozessor Xalan von der Apache Software Foundation (<http://xalan.apache.org/>) transformiert XML-Dokumente in HTML-, Klartext- oder andere XML-Dokumente mithilfe von XSLT-Stylesheets. Xerces ermöglicht dabei das Parsen, Modifizieren und Generieren von XML-Daten sowie den Aufbau einer objektorientierten Darstellung von XML-Dokumenten.

Zur Ausführung des VokaDoks-Designers werden beide Archiv-Dateien in das Verzeichnis `\lib\ext\` des Installationsverzeichnisses des JRE kopiert.

#### F.1.5.4 servlet-api.jar

Die `servlet-api.jar`, welche im Ordner `\lib\` des Installationsverzeichnisses des Webcontainers Tomcat zu finden ist, dient der Ausführung von Servlets. Servlets sind Java-Klassen, die Anfragen von Clients

zum Webserver weiterleiten und die gelieferten Antworten zur Darstellung schicken. Die `servlet-api.jar` soll dem Klassenpfad des Systems hinzugefügt werden.

Für die Ausführung des VokaDoks-Designers sollte ggf. die `servlet-api.jar` in JRE kopiert werden (z.B. in das Verzeichnis `\lib\ext\` des Installationsverzeichnis des JRE). Sie sollte jedoch nach der Modellierung mit dem Designer aus dem JRE gelöscht werden, damit bei der Ausführung der Webanwendung für das Dokumentationssystem kein Nutzungskonflikt entsteht.

#### F.1.5.5 Spezifikation zum genutzten MySQL

Für die Verwaltung, Nutzung und Pflege von kontrollierten Vokabularen, die im VokaDoks-Designer Anwendung finden, wurde eine MySQL-Datenbank in der Version 5.5 erstellt. In dieser Vokabular-Datenbank befinden sich ICD-10-GM 2010, OPS 2010, TNM 2006, MedDRA v.13.0, und SNOMED-CT 2010 als standardisierte kontrollierte Vokabulare in der Medizin. Der Zugriff auf diese Objektrelationale Datenbank erfolgt mit dem Tool MySQL Workbench 5.2 CE und zum Abfragen und Manipulieren der Daten wird auf die Datenbanksprache SQL (Structured Query Language) zurückgegriffen. Die Installation dieser Vokabular-Datenbank ist Voraussetzung für die Nutzung des *Vocabularies Systems* des VokaDoks-Designers zur semantischen Beschreibung von Merkmalen bzw. zur Erzeugung von standardisierten Stammdaten.

Eine zweite MySQL-Datenbank wird außerdem zur Speicherung von den im generierten VokaDoks-Dokumentationssystem eingegebenen Daten benötigt. Zur Erzeugung einer SQL-Datei aus dem VokaDoks-Designer, die zur Generierung einer Dokumentationssystem-Datenbank dienen wird, benötigt der VokaDoks-Designer einen JDBC-Treiber. Der JDBC-Treiber `mysql-connector-java-5.1.22-bin.jar` wurde in das Verzeichnis `\lib\ext\` des Installationsverzeichnis des JRE kopiert und dem Klassenpfad des Systems hinzugefügt. Zum Beispiel: `classpath = C:\Programme\Java\jre7\lib\ext\mysql-connector-java-5.1.22-bin.jar`. Nach der Modellierung mit dem VokaDoks-Designer und vor Ausführung der CDMS-Anwendung muss der JDBC-Treiber ausgeschnitten und ins Webcontainer-Installationsverzeichnis kopiert werden (bei Anwendung von Tomcat als Webcontainer wäre es zum Beispiel unter `C:\Program Files\Tomcat 7.0\lib`).

#### F.1.5.6 Spezifikation zum genutzten Webcontainer

Die CDMS-Anwendung ist eine Webanwendung und benötigt zum Laufen einen Webcontainer. Tomcat ist ein Webcontainer, der die Ausführung von in Java geschriebenen Webanwendungen auf Servlet-Basis ermöglicht. Somit können Abfragen von Clients verarbeitet und beantwortet werden. Tomcat muss als Webserver vom Betriebssystem identifiziert werden, deshalb sollte er dem Systempfad (Umgebungsvariablen) hinzugefügt werden. Zum Beispiel: `path=C:\Programme\Tomcat 7.0\bin` und `CATALINA_HOME=C:\Programme\Tomcat 7.0`. Zum Laufen setzt Tomcat die Installation von Java voraus. Deshalb empfiehlt sich das Java-Archiv `tools.jar` aus dem Java Verzeichnis (z.B.: `C:\Programme\Java\jdk1.7.0_51\lib`) in den Ordner `\lib\` des Tomcat

Installationsverzeichnis zu kopieren. Die Werkzeug-Bibliothek tools.jar ist ein http-Server, der das Herunterladen von .class- und .jar-Dateien zulässt.

### **F.1.6 Einschränkungen**

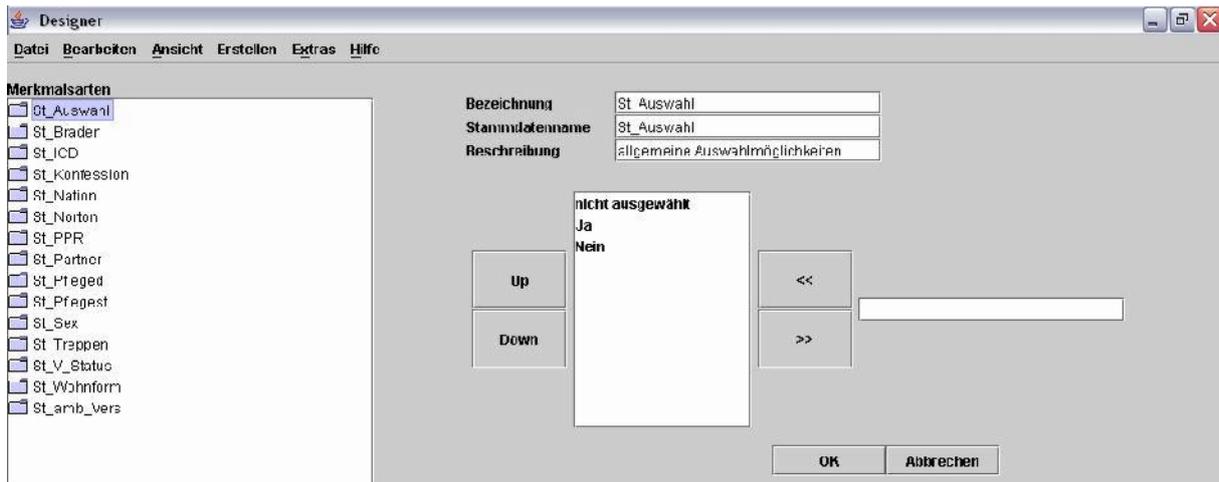
Die Einbindung des Moduls *Clinical Thesaurus* des VokaDoks-Dokumentationsmodells wird in den Ausbau des InCoMe-Designers nicht umgesetzt. Der Schwerpunkt bei diesem Ausbau wurde auf die Umsetzung der Hauptmodule *Vocabularies System* und *Clinical Documentation System* des VokaDoks gelegt, da sie die Kernkomponenten eines terminologisch fundierten Dokumentationssystems darstellen und für die Verwaltung von kontrollierten Vokabularen, sowie die Modellierung von Dokumentationsmerkmalen und von Wertebereichen für Dokumentationsmerkmale essenziell sind. Außerdem lassen sich InCoMe-Modellelemente ziemlich gut und leicht in diesen beiden VokaDoks-Hauptmodulen abbilden.

Um die Vokabular-Datenbank mit ICD-10-GM 2010, OPS 2010, TNM 2006, MedDRA v.13.0, und SNOMED-CT 2010 im VokaDoks-Designer außerhalb dem wissenschaftlichen Rahmen benutzen zu dürfen, muss sich jeder VokaDoks-Benutzer entsprechende Lizenzen zur Nutzung jedes Vokabulars erwerben. Die Datenbank-Zugangsdaten sind statisch und sollten nicht geändert werden, um das Vokabular-Modul des VokaDoks-Designers einwandfrei nutzen zu können. Tabelle 5 führt die MySQL-Zugangsdaten für die VokaDoks-Vokabular-Datenbank auf, so wie sie einzurichten sind. Eine Änderung dieser Datenbank-Parameter bedeutet eine Anpassung des VokaDoks-Designers in seiner Implementierung.

**Tabelle 5: Zugangsdaten der VokaDoks-Vokabular-Datenbank**

<b>Datenbank-Parameter</b>	<b>Wert</b>
USERNAME	root
PASS	motsecret
DATENBANKNAME	iso11179v3
MySQL Portnummer	3306
JDBC_DRIVER	com.mysql.jdbc.Driver
DB_URL	jdbc:mysql://localhost:3306/





**Abbildung 24: Ansicht zur Erfassung der benötigten Stammdaten am Beispiel allgemeine Auswahlmöglichkeiten [20]**

## F.2 SYSTEMBESCHREIBUNG

### F.2.1 Einführung

Der Abschnitt Systembeschreibung dient der Beschreibung des Ablaufs über den ein Dokumentar/Anwender den VokaDoks-Designer nutzen kann, um ein Terminologie-basiertes Dokumentationssystem zu spezifizieren. Dieser Abschnitt könnte daher als kleines Anwenderhandbuch für den VokaDoks-Designer gesehen werden. Die Ergebnisse der Implementierung des VokaDoks-Modells werden auch hier vorgestellt, indem das resultierende System mit seinen Komponenten beschrieben wird. Diese Implementierung basiert auf den Ergebnissen aus dem Ausbau von InCoMe mit VokaDoks-Komponenten.

Das Dokumentationsmodell besteht aus den Elementen: Merkmalsarten (vgl. Data\_Element), Stammdaten (vgl. Value\_Domain), Kontexten (vgl. Data\_Element\_Concept) und Datenbank (vgl. Concept\_System) sowie ihren entsprechenden Integritätsbedingungen, die sich über eine graphische Benutzeroberfläche – den VokaDoks-Designer – erfassen, bearbeiten und zusammenstellen lassen. Entsprechend diesen Elementen gibt es unterschiedliche Ansichten bzw. Editoren: Merkmalsart, Kontext, Stammdaten. Um ein bestimmtes Element erzeugen zu können, muss man deshalb vorab zu dem entsprechenden Editor wechseln.

Der VokaDoks-Designer (designer.jar) lässt sich über ein DOS-Eingabefenster durch Eingabe der Kommando-Zeile `Java -jar .designer.jar` oder durch einfachen Doppelklick auf die Datei `designer.jar` starten. Abschnitt F.1.5 beinhaltet alle nützlichen Informationen zur Überprüfung aller benötigten Systemeinstellungen zur einwandfreien Ausführung des VokaDoks-Designers.

### F.2.2 Nutzung von kontrollierten Vokabularen

Der Datenbestand von VokaDoks besteht aus insgesamt 6 unterschiedlichen medizinischen kontrollierten Vokabularen: ICD-10-GM 2010, OPS 2010, TNM 2006, MedDRA v.13.0 und

SNOMED-CT 2010. Die umfangreiche Nutzung von kontrollierten Vokabularen wird bei der Erfassung von Wertebereichen sichtbar (Abschnitt F.2.4).

### F.2.3 Erfassung von Merkmalsarten/Data Elements (Merkmalsebene)

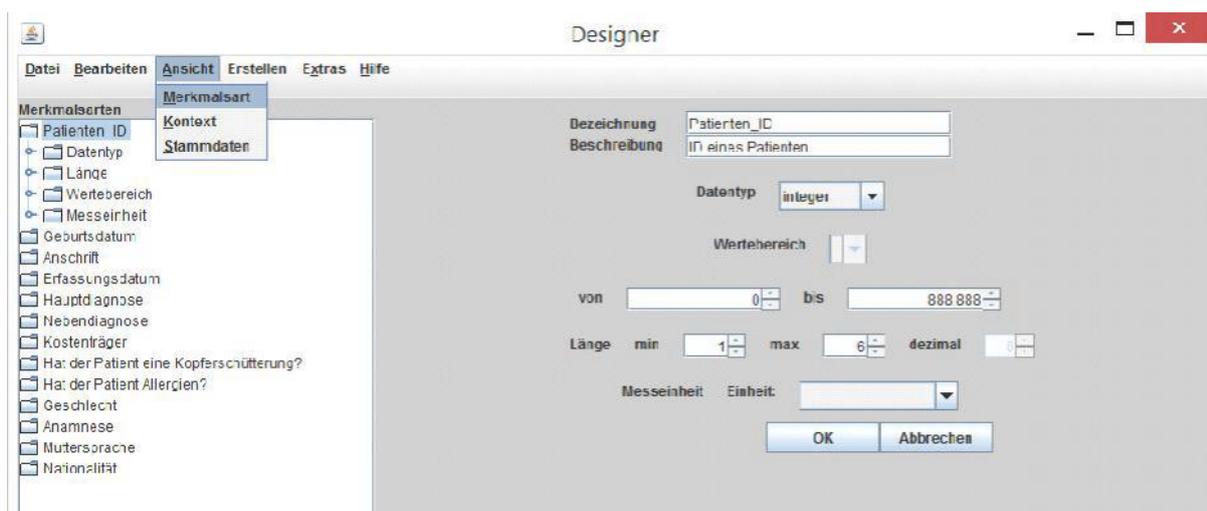
#### F.2.3.1 Einführung

Um den Merkmalskatalog eines Dokumentationssystems bzw. eines Registers erstellen zu können, müssen Dokumentationseinheiten definiert und ihre Integritätsbedingungen festgelegt werden. Dies geschieht durch den Editor Merkmalsart unter dem Menüeintrag Ansicht → Merkmalsart und Bearbeiten → Merkmalsart hinzufügen (s. Abbildung 25). Dabei können die Integritätsbedingungen Datentyp, Datenlänge (bei Zahlen und Zeichenketten), Wertebereich (bei Zahlen) und Messeinheit (bei Zahlen) modelliert werden.

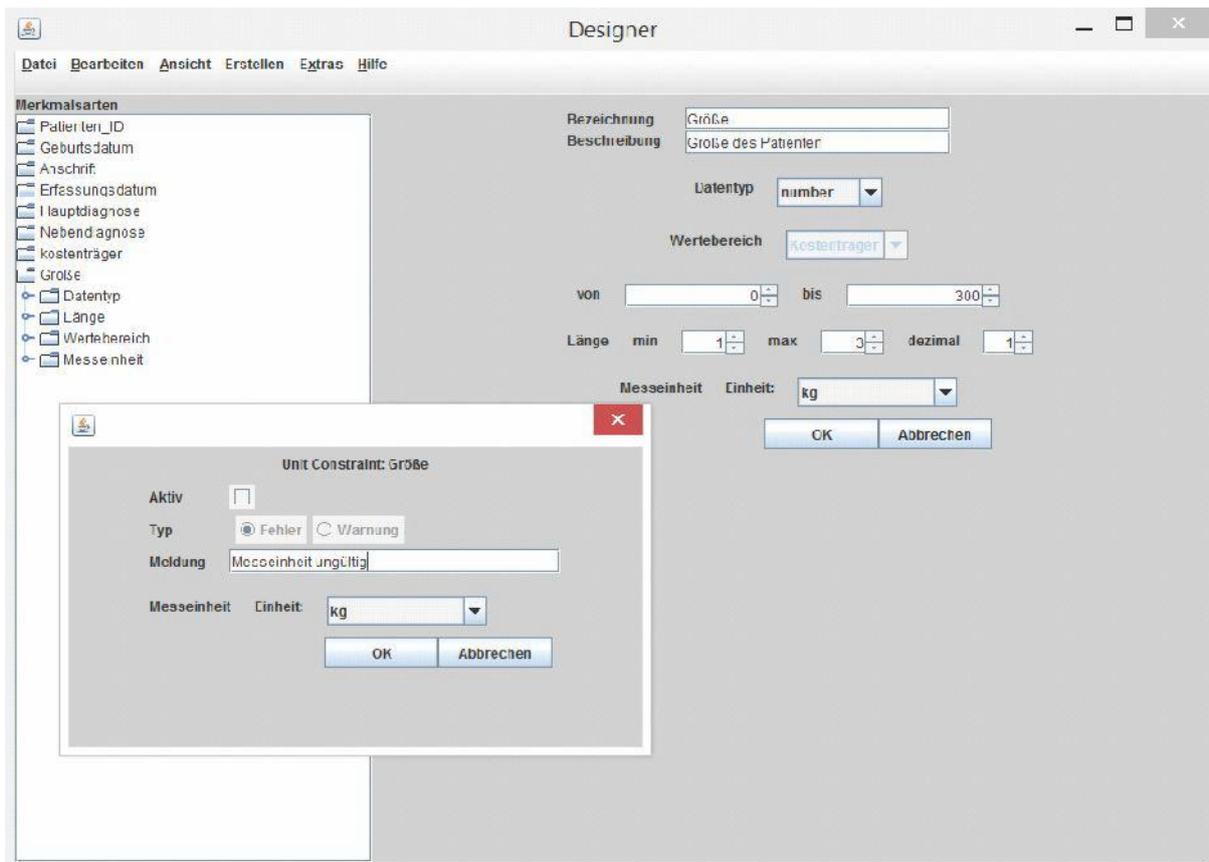
Für Merkmalsarten, die durch Stammdaten eingeschränkt werden (Datentyp = „reference“), muss als Wertebereich eine vordefinierte Werteliste ausgewählt werden. Abbildung 28 zeigt die Zuweisung von Stammdaten „Kostenträger“ zur gleichnamigen Merkmalsart „Kostenträger“.

#### F.2.3.2 Erfassung von Integritätsbedingungen für Merkmalsarten

Nach Erfassung von Merkmalsarten können ihre definierten Integritätsbedingungen einzeln überarbeitet werden. Durch Doppelklick auf eine Merkmalsart öffnet sich die Liste ihrer Integritätsbedingungen und durch einfachen Klick auf eine Integritätsbedingung gelangt man zu einer internen graphischen Oberfläche, in der die Integritätsbedingung überarbeitet werden kann. Abbildung 26 zeigt als Beispiel die Überarbeitung der Integritätsbedingung für die Messeinheit für die Merkmalsart Größe. Hier kann die Fehlermeldung bei einer falschen Eingabe verfasst, sowie die Messeinheit für die Merkmalsart umgeändert werden. Die Integritätsbedingung wird dem Benutzer als Fehler angezeigt (voreingestellt für Merkmale vom Datentyp Integer).



**Abbildung 25: Erfassung von Merkmalsarten. Eine Patienten\_ID darf höchstens sechs-stellig bis zum Maximalwert 888.888 sein.**



**Abbildung 26: Überarbeitung von Integritätsbedingungen von Merkmalsarten**

#### ***F.2.4 Erfassung von Stammdaten (Wertebereiche)***

Hier gibt es die Möglichkeit auf das *Vocabularies System* zuzugreifen, um standardisierte medizinische Begriffe als Stammdaten auszuwählen sowie medizinische präkoordinierte Datensätze (*Statement*) zusammenzustellen. Es ist ebenfalls möglich nicht standardisierte medizinische Begriffe manuell einzugeben (s. Abbildung 27).

Unter dem Menüeintrag *Ansicht* → *Stammdaten*, und *Bearbeiten* → *Stammdaten hinzufügen* können Stammdaten definiert werden.

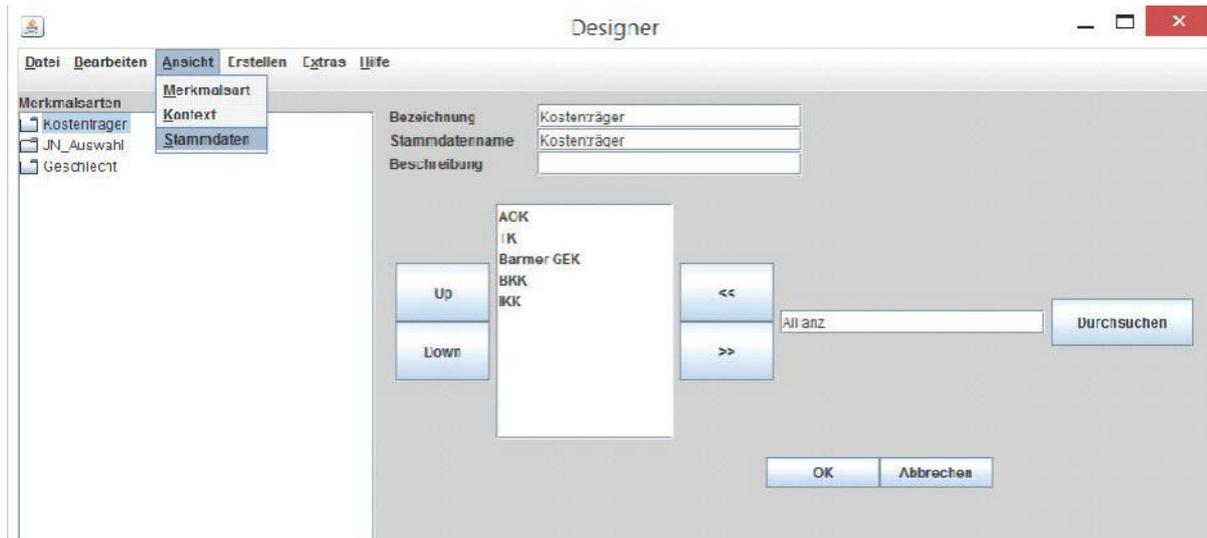


Abbildung 27: Erfassung von Stammdaten durch manuelle Eingabe

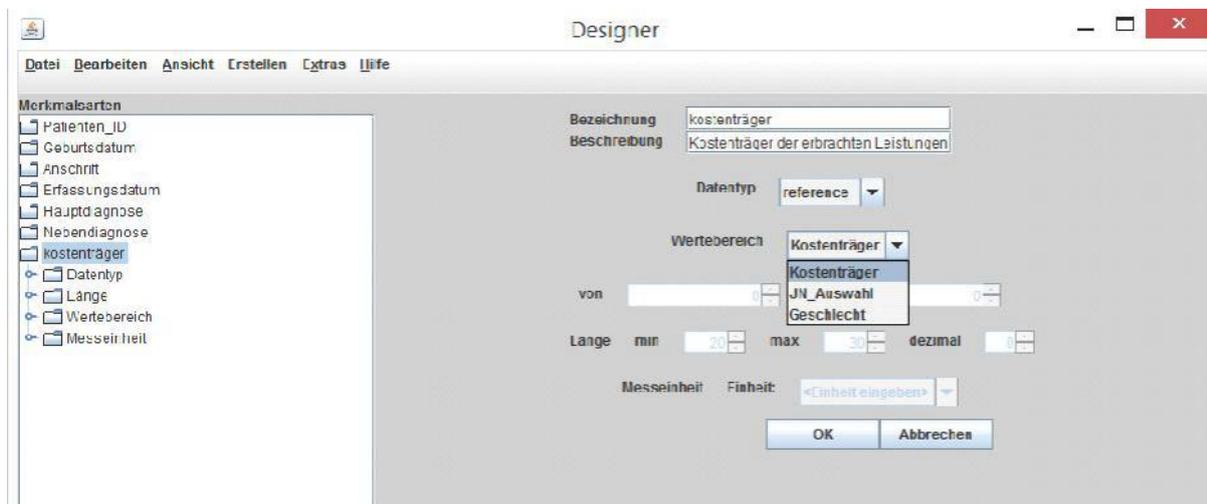


Abbildung 28: Zuweisung von Stammdaten

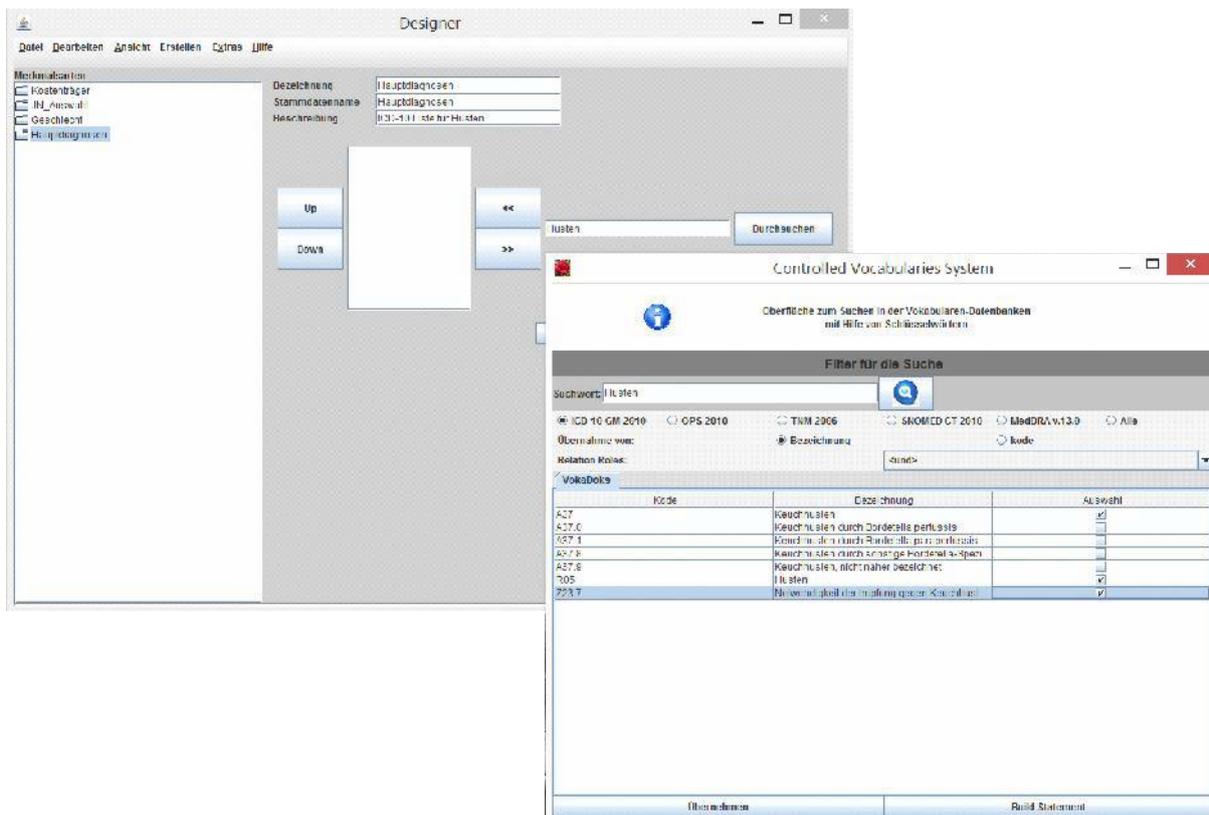
#### F.2.4.1 Erfassung von standardisierten medizinischen Begriffen

Für Merkmalsarten, die durch standardisierte medizinische Begriffe eingeschränkt werden, gibt es die Möglichkeit nach standardisierten Konzepten (Kode und Bezeichnung) im *Vocabularies System* zu suchen. Dies erfolgt im Editor Stammdaten durch den Knopf „Durchsuchen“. Es gibt die Möglichkeit einzeln in ICD-10-GM 2010, OPS 2010, TNM 2006, SNOMED-CT 2010, MedDRA v.13.0 oder in allen diesen Vokabularen nach standardisierten Begriffen zu suchen. In der Ergebnistabelle dürfen diejenigen Begriffe bzw. Codes ausgewählt werden, die einer spezifischen Merkmalsart entsprechen. Für Stammdaten können Codes oder (exklusiv) Bezeichnungen ausgewählt werden. Abbildung 29 zeigt zum Beispiel die Auswahl von ICD-10-GM Bezeichnungen für Husten für die Merkmalsart Hauptdiagnose. Durch einfachen Klick auf den Knopf „Übernehmen“ wird die Auswahlliste als Stammdaten übernommen (s. Abbildung 31).

Es gibt die Möglichkeit seine Auswahlliste zu verbessern bzw. zu verfeinern indem man andere kontrollierte Vokabulare nach demselben oder anderen Begriffen durchsucht. Abbildung 30 zeigt als Beispiel die Ergebnistabelle für den Begriff „Husten“ in MedDRA v.13.0.

Nach Übernahme der standardisierten Auswahlliste besteht weiterhin die Möglichkeit nicht standardisierte Begriffe manuell einzugeben (s. Abbildung 31).

Die Erstellung der Stammdaten für die Hauptdiagnose wird durch Betätigung des OK-Knopfes fertiggestellt.



**Abbildung 29: Durchsuchen von standardisierten medizinischen Konzepten im Vokabularsystem. Der Begriff „Husten“ wird im ICD-10-GM 2010 gesucht und passende Hauptdiagnosen ausgewählt.**

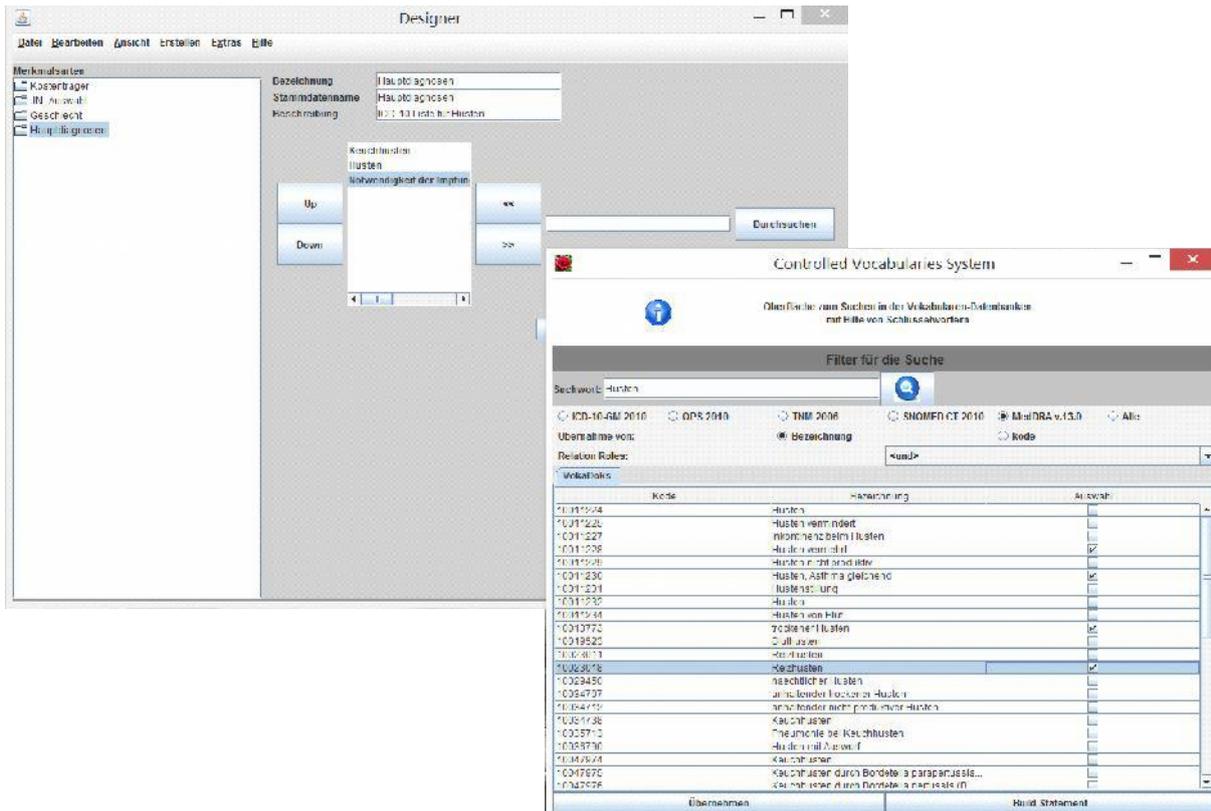


Abbildung 30: Verfeinerung von Stammdaten durch Suchen in anderen medizinischen Vokabularen. Der Begriff „Husten“ wird in MedDRA v.13.0 gesucht.



Abbildung 31: Übernahme von standardisierten medizinischen Begriffen und Möglichkeit der zusätzlichen Eingabe nicht standardisierter Begriffe

#### F.2.4.2 Erfassung von medizinischen präkoordinierten Datensätzen

Für die Erfassung von medizinischen präkoordinierten Datensätzen werden Begriffe in einzelnen oder in allen vorhandenen Vokabularen gesucht, die Übernahme von Code oder Bezeichnung ausgewählt, eine Beziehungsrolle (Relation Roles) ausgewählt und jeweils nur zwei Konzepte ausgewählt. Durch

einfachen Klick auf den Knopf „Build Statement“ werden die präkoordinierten Bezeichnungen von Konzepten in die Stammdaten übernommen (s. Abbildung 32).



**Abbildung 32: Übernahme von präkoordinierten Bezeichnungen von medizinischen Konzepten in die Stammdaten**

#### *F.2.5 Bündelung von Kontexten/Data\_Element\_Concept*

Vor der Gruppierung von Kontexten (vgl. Data\_Element\_Concept: Zusammensetzung von Data\_Elements) muss eine Datenbank (vgl. Concept\_System: Zusammensetzung von Concepts) definiert und somit auch der medizinische Kontext des modellierten Merkmalskatalogs festgelegt werden. Unter dem Menüeintrag Ansicht → Kontext, und Bearbeiten → Kontext hinzufügen können Datenbank und Kontexte angelegt werden (s. Abbildung 33). Anschließend werden die Kontexte angelegt, in dem Merkmalsarten ihrem entsprechenden Kontext zugeordnet werden. Goertzen nennt solche Merkmalsarten, die Kontexte zugeordnet werden „gebundene Merkmalsarten“, da sie zu einem bestimmten Kontext gebunden werden und somit ihre volle Semantik in dem medizinischen Kontext erhalten.

Nach Erscheinen der Ansicht zum Anlegen von Kontexten klickt man zunächst auf die Wurzel (Stamm) des Baumes (s. roter Pfeil in Abbildung 33). Dies ermöglicht die Datenbank anzulegen (s. Abbildung 34). Danach klickt man auf das zweite Kindelement der angelegten Datenbank (s. roter Pfeil in Abbildung 34), um den ersten Kontext anzulegen und seine gebundenen Merkmalsarten festzulegen, indem sie aus der rechten Merkmalsartenliste ausgewählt und in die linke Merkmalsartenliste für den Kontext hinzugefügt werden (s. Abbildung 35). Weitere Kontexte werden durch den Menüpunkt Bearbeiten → Kontext hinzufügen hinzugefügt.

Zusätzlich zu der Bündelung von Kontexten sind spezifische Integritätsbedingungen zu definieren. Jede zu einem Kontext gebundene Merkmalsart besitzt Schlüssel-, Anzahl-, Referenz (Fremdschlüssel)-, Suche- und Ergebnis-Integritätsbedingung (s. Abschnitt C.3.2).

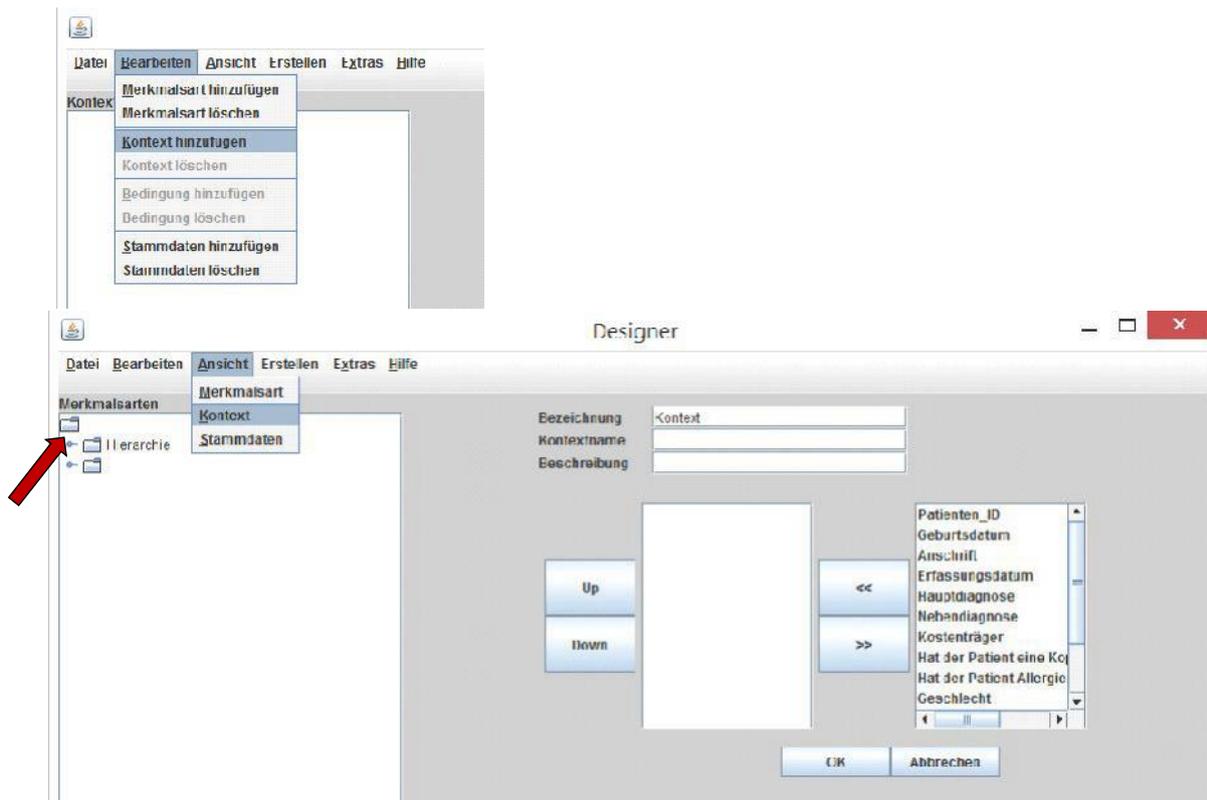
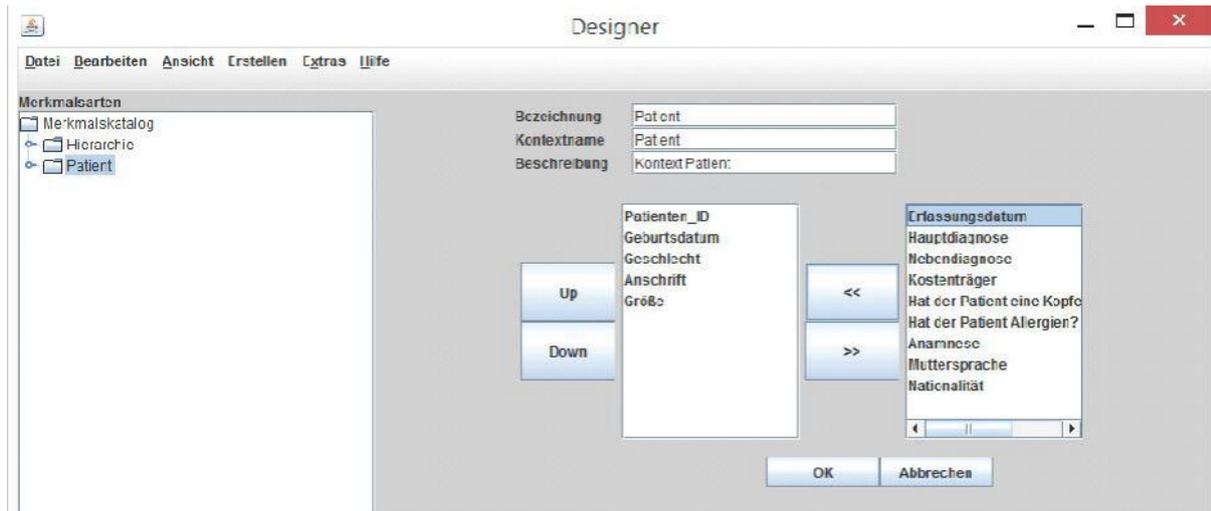


Abbildung 33: Bereich der Anwendung: Startansicht zum Anlegen von Datenbank und Kontexten



Abbildung 34: Ansicht zum Anlegen einer Datenbank



**Abbildung 35: Ansicht zum Anlegen eines Kontexts**

### ***F.2.6 Definition von Integritätsbedingungen (Constraints)***

#### **F.2.6.1 Hierarchie**

Die Hierarchie der Kontexte innerhalb einer Datenbank kann festgelegt werden. Dies dient der Festlegung der hierarchischen Struktur des Dokumentationssystems. Somit wird die Navigation zwischen den Kontexten in der CDMS-Anwendung erleichtert. Im Beispiel befindet sich auf der obersten Hierarchie-Ebene der Kontext Patient. In der CDMS-Anwendung wird deshalb zuerst auf das Modul Patient zugegriffen, um die patientenbezogenen Merkmale zu erfassen. Ausgehend vom Kontext Patient besteht die Möglichkeit Daten für die Module Visite und Sozialdaten zu erfassen (s. Abbildung 36). Jeder Kontext besitzt ebenfalls eine Hierarchie-Integritätsbedingung (s. Abbildung 37).

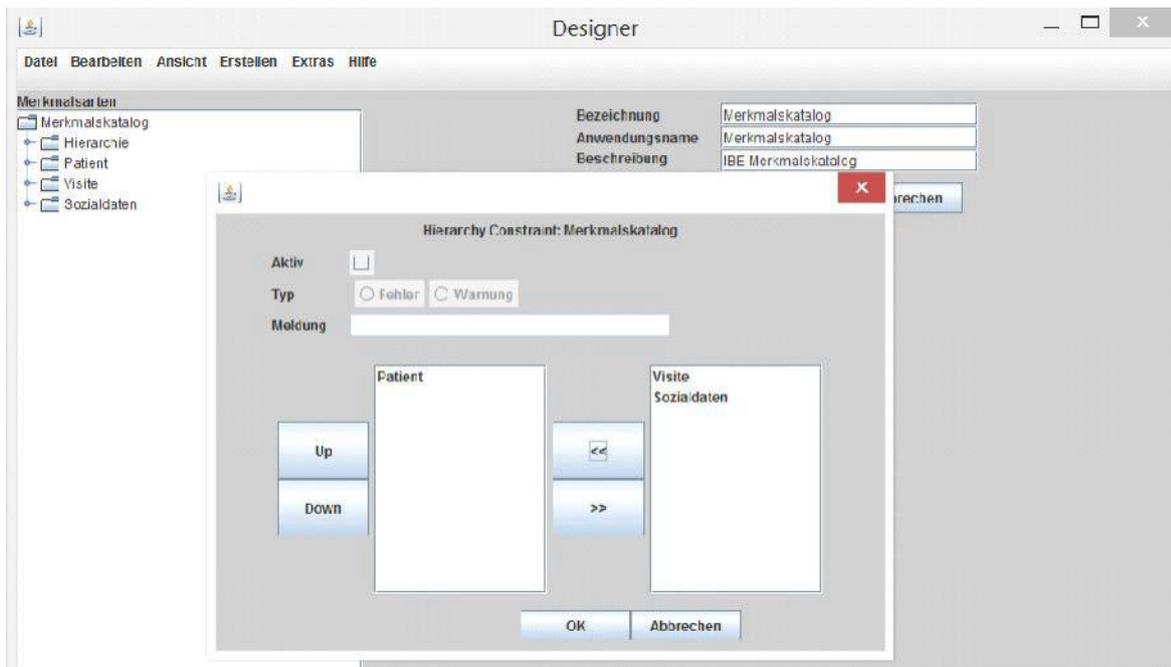


Abbildung 36: Hierarchie-Integritätsbedingung für die Datenbank Merkmalskatalog

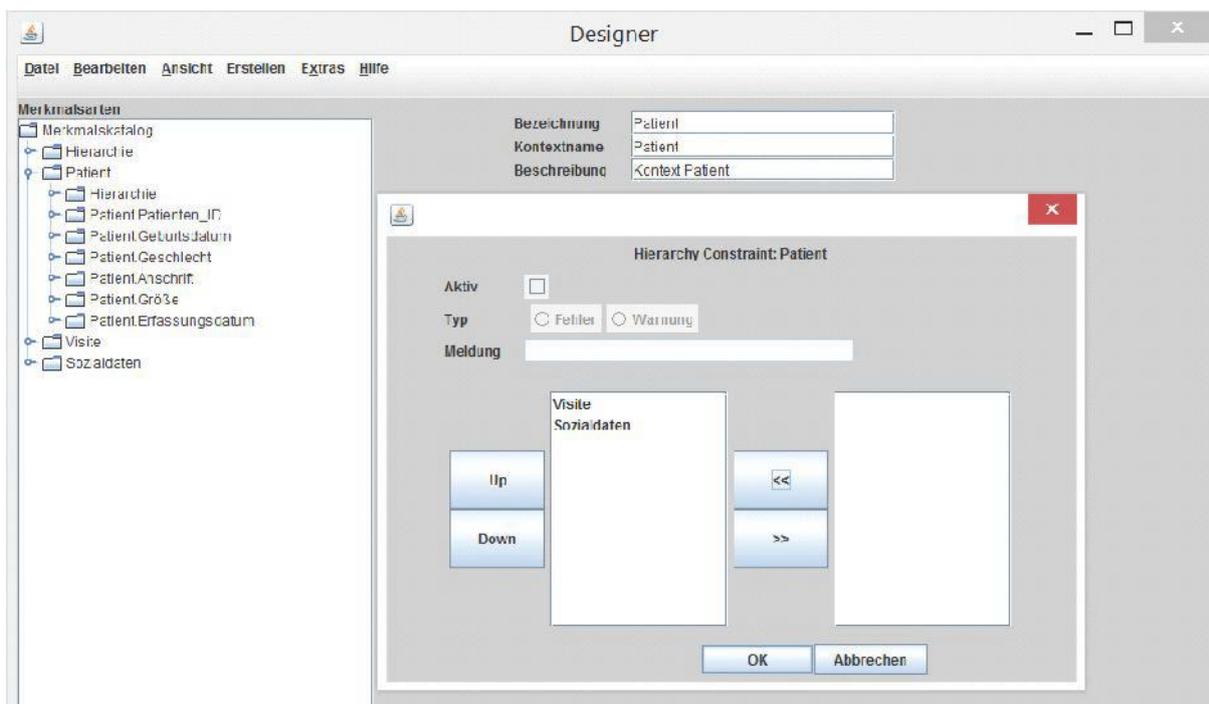
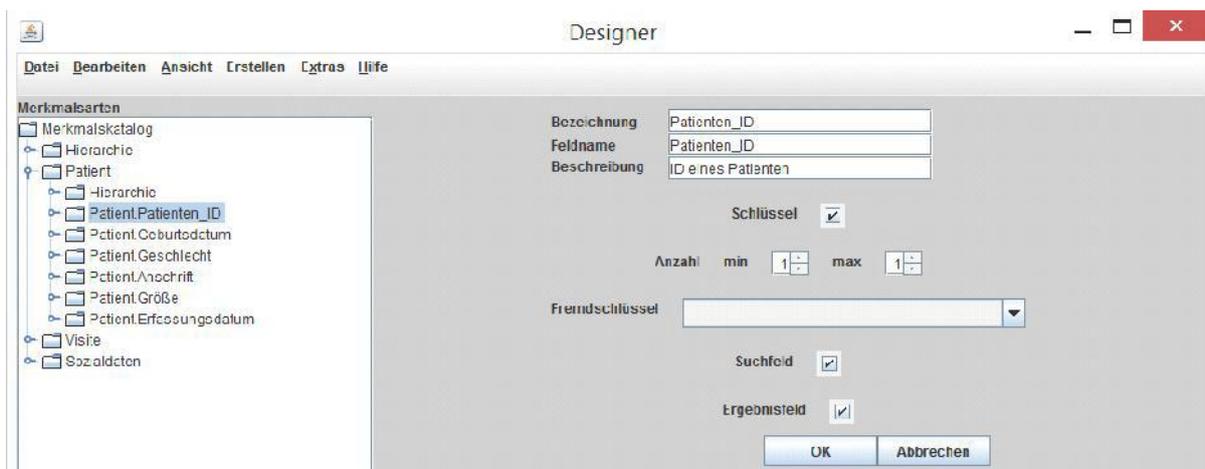


Abbildung 37: Hierarchie-Integritätsbedingung für den Kontext Patient

### F.2.6.2 Schlüssel

Jeder Kontext soll über einen Primärschlüssel (Primary Key) eindeutig identifizierbar sein. Für den Kontext Patient ist zum Beispiel der Primärschlüssel die eindeutige ID eines Patienten (gebundene Merkmalsart: Patienten\_ID) (s. Abbildung 38). Für die Kontexte Visite und Sozialdaten soll der Primärschlüssel durch ein Tupel aus Patienten\_ID (empfohlen aber optional, da bereits

Fremdschlüssel, s. F.2.6.4) und mindestens einer anderen Merkmalsart gebildet werden. Als Beispiel könnte man hier für den Kontext Visite das Tupel (Patienten\_ID, Erfassungsdatum) als Primärschlüssel zur eindeutigen Identifizierung einer Patientenvisite definieren (Häkchen beim Feld Schlüssel setzen) oder eine ganz neue Merkmalsart zu diesem Zweck modellieren (z.B. eine allgemeine ID). Diese Modellierung macht sich vor allem auf der Anwendungsebene bemerkbar. Es wird nämlich dadurch ermöglicht, zu einem Patienten mehrere Visiten anlegen zu können. Würde man aber zum Kontext Visite keinen eigenen bzw. eindeutigen Primärschlüssel definieren, würde man für einen Patienten auf der Anwendungsebene lediglich eine einzige Visite anlegen können. Somit hält sich die Modellierung an die Normalisierungsregeln eines relationalen Datenschemas.



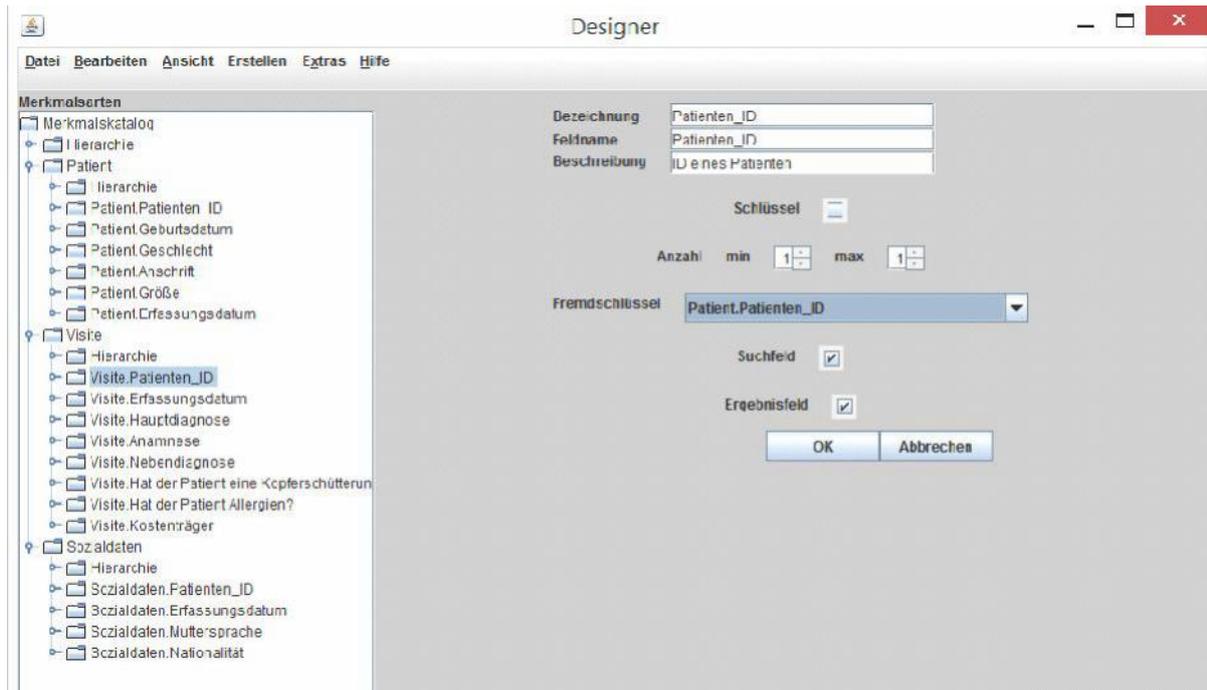
**Abbildung 38: Schlüssel-Integritätsbedingung: Ein Patient wird durch seine Patienten\_ID eindeutig identifiziert.**

### F.2.6.3 Anzahl

Jede zu einem Kontext gebundene Merkmalsart ist für die Erfassung entweder Pflicht (min=1 und max=1) oder optional (min=0 und max=1) oder irrelevant (min=0 und max=0). In Abbildung 38 ist die Eingabe der Patienten\_ID Pflicht.

### F.2.6.4 Referenz (Fremdschlüssel)

Es gibt gebundene Merkmalsarten eines bestimmten Kontexts, deren Eingabe für die Semantik anderer Kontexte unerlässlich ist. Solche Merkmalsarten werden von den anderen Kontexten referenziert (s. Abbildung 39).



**Abbildung 39: Referenz-Integritätsbedingung: die Patienten\_ID in den Kontexten Visite und Sozialdaten referenziert jeweils die Patienten\_ID des Kontexts Patient.**

#### F.2.6.5 Suche

Durch die Suche-Integritätsbedingung wird festgelegt, ob es sich bei der Merkmalsart um ein Suchfeld handelt. Somit kann man einen Patienten anhand seiner Patienten\_ID, seines Geburtsdatum oder des Erfassungsdatums suchen.

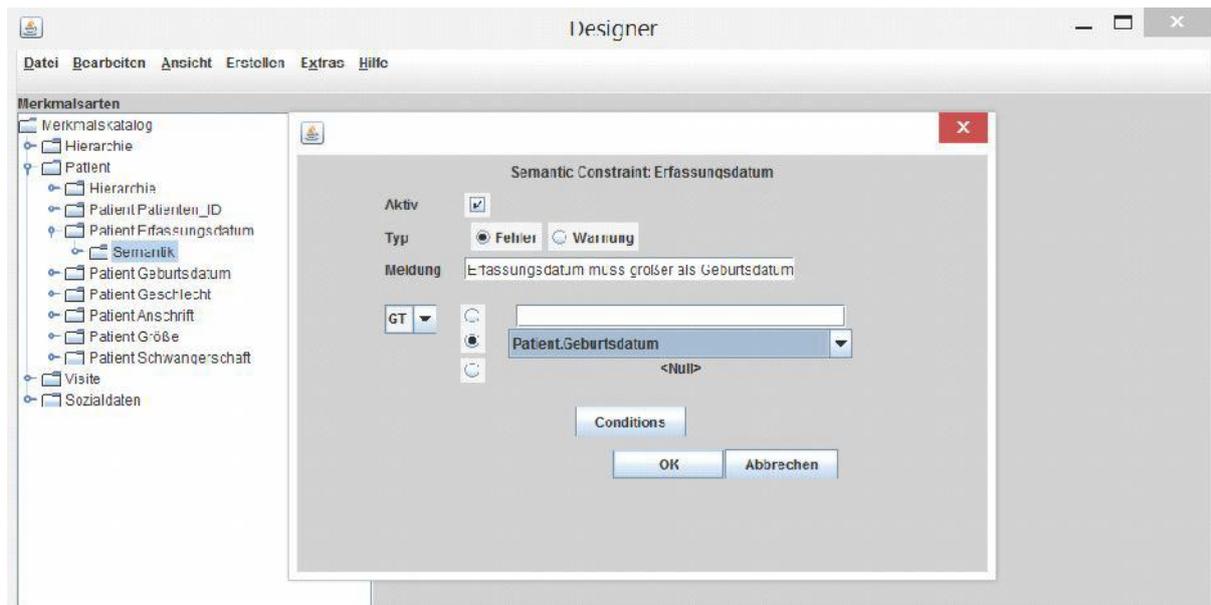
#### F.2.6.6 Ergebnis

Durch die Ergebnis-Integritätsbedingung wird festgelegt, ob eine Merkmalsart nach einer Suche in der Ergebnis-Liste angezeigt werden soll. Nach der Suche nach einem Patienten werden seine Patienten\_ID, sein Geburtsdatum, sein Geschlecht, seine Größe und das Erfassungsdatum angezeigt.

#### F.2.6.7 Semantische Integritätsbedingung

Durch die semantische Integritätsbedingung werden Regeln über semantische Eigenschaften zusammengehöriger Merkmalsarten festgelegt. Diese Regeln werden durch logische Ausdrücke definiert. Semantische Integritätsbedingungen lassen sich definieren, in dem auf einem Kontext oder einer gebundenen Merkmalsart geklickt und anschließend der Menüeintrag Bearbeiten → Bedingung hinzufügen gewählt wird. Bei der Modellierung der semantischen Integritätsbedingung kann über den Status (Aktiv: Soll die Meldung in dem späteren CDMS angezeigt werden?), den Meldungstyp (Typ: Handelt es sich bei der Meldung um einen Fehler oder eine Warnung) und den anzuzeigenden Meldungstext der Integritätsbedingung entschieden werden.

Die gebundene Merkmalsart Erfassungsdatum muss zum Beispiel größer sein (GT: Greater Than) als das Geburtsdatum eines Patienten (s. Abbildung 40).

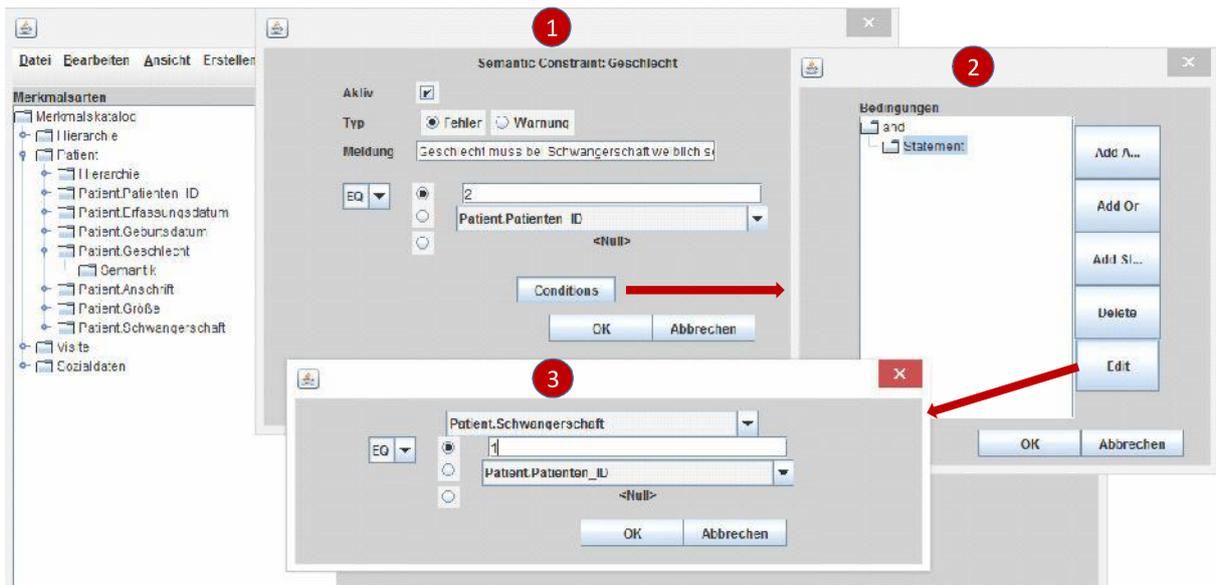


**Abbildung 40: Semantische Integritätsbedingung: Das Erfassungsdatum muss größer sein als das Geburtsdatum eines Patienten.**

Ein weiteres Beispiel wäre das Vorliegen einer Schwangerschaft nur bei weiblichen Patienten. Von der geltenden semantische Aussage „Nur weibliche Patienten können schwanger sein“ wird die entsprechende zu modellierende Integritätsbedingung „Die Merkmalsart Geschlecht im Kontext Patient muss die Ausprägung ‚weiblich‘ haben, damit die Merkmalsart Schwangerschaft die Ausprägung ‚ja‘ bekommt“ abgeleitet. Die Datenbank-Kodierung der Ausprägung ‚weiblich‘ in den Stammdaten Geschlecht {männlich, weiblich} ist 2 (zweite Position in der Liste). Die Datenbank-Kodierung der Ausprägung ‚ja‘ in den Stammdaten Ja\_Nein\_Unbekannt {ja, nein, unbekannt} ist 1 (erste Position in der Liste). Nach Erfassung der ersten Aussage („Geschlecht ist weiblich“: <Geschlecht EQ 2>) erfolgt durch Klicken auf den Knopf „Conditions“ die Erfassung der Bedingung („Schwangerschaft ist ja“: <Schwangerschaft EQ 1>), in dem unter dem logischen Operator „and“ ein Statement hinzugefügt („Add Statement“) und durch Klicken auf den Knopf „Edit“ erfasst wird (s. Abbildung 41).

Übernimmt man durch Referenzierung (s. F.2.6.4) die gebundene Merkmalsart eines Kontexts in einem anderen Kontext, müssen alle Integritätsbedingungen für diese Referenz erneut modelliert werden. Würde man zum Beispiel die gebundene Merkmalsart „Schwangerschaft“ des Kontexts Patient im Kontext Visite referenzieren, müsste man die oben beschriebene semantische Integritätsbedingung erneut modellieren:

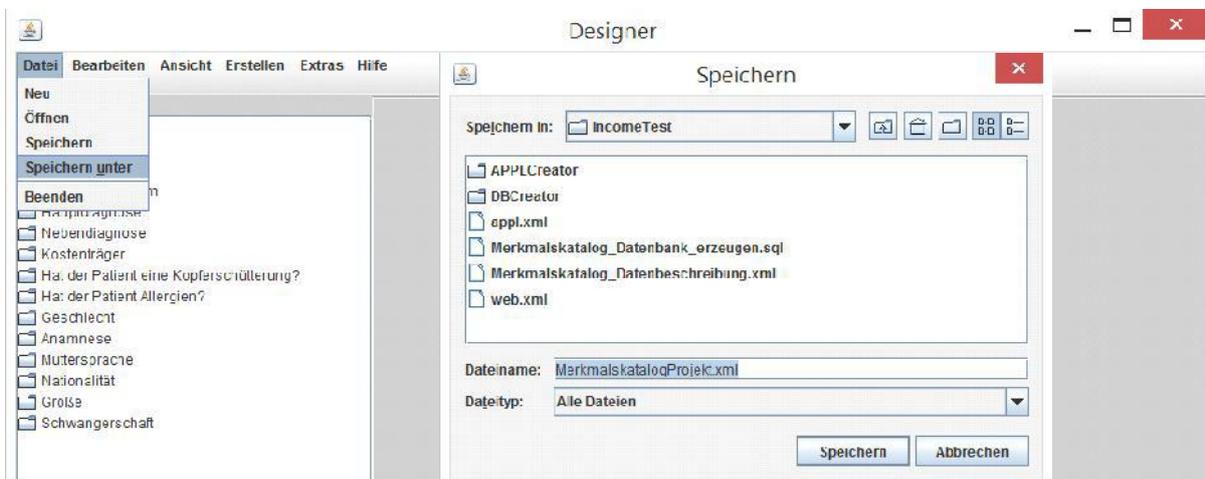
(„Schwangerschaft ist ja“: <Visite.Schwangerschaft EQ 1>) nur wenn („Geschlecht ist weiblich“: <Patient.Geschlecht EQ 2>).



**Abbildung 41: Semantische Integritätsbedingung: Nur weibliche Patienten können schwanger sein.**

### F.2.7 Sicherung der Modellierung

Der Entwurf des Dokumentationsmodells lässt sich als XML-Datei speichern und bei Bedarf mit dem Designer wiederöffnen, um zum Beispiel Änderungen an den modellierten Daten vorzunehmen (s. Abbildung 42).



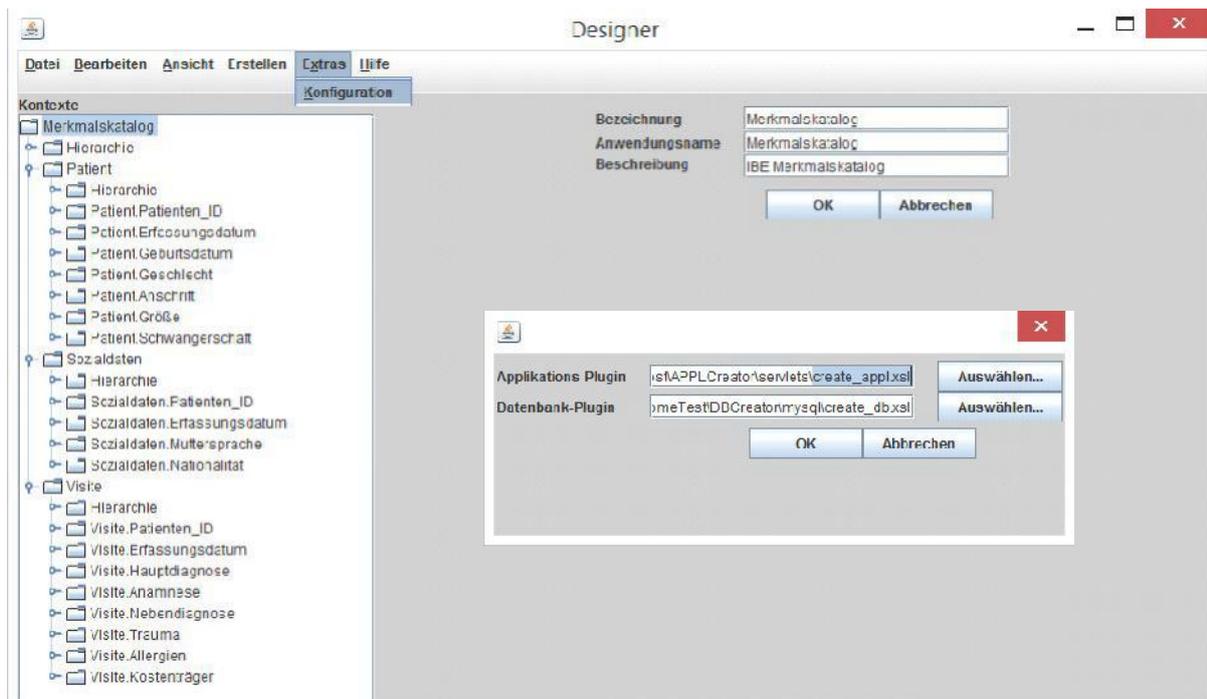
**Abbildung 42: Sicherung des Entwurfsprojekts mit dem VokaDoks-Designer**

### F.2.8 Export des Dokumentationsmodells

#### F.2.8.1 Einführung

Das Generierungsmodul wird in diesem Teil benötigt, um das Dokumentationsmodell bzw. den entworfenen Merkmalskatalog in eine Datenbeschreibung in SQL und eine Anwendungsbeschreibung in XML zu transformieren. Die Transformationsschnittstellen `create_db.xml` und `create_appl.xml` (eine

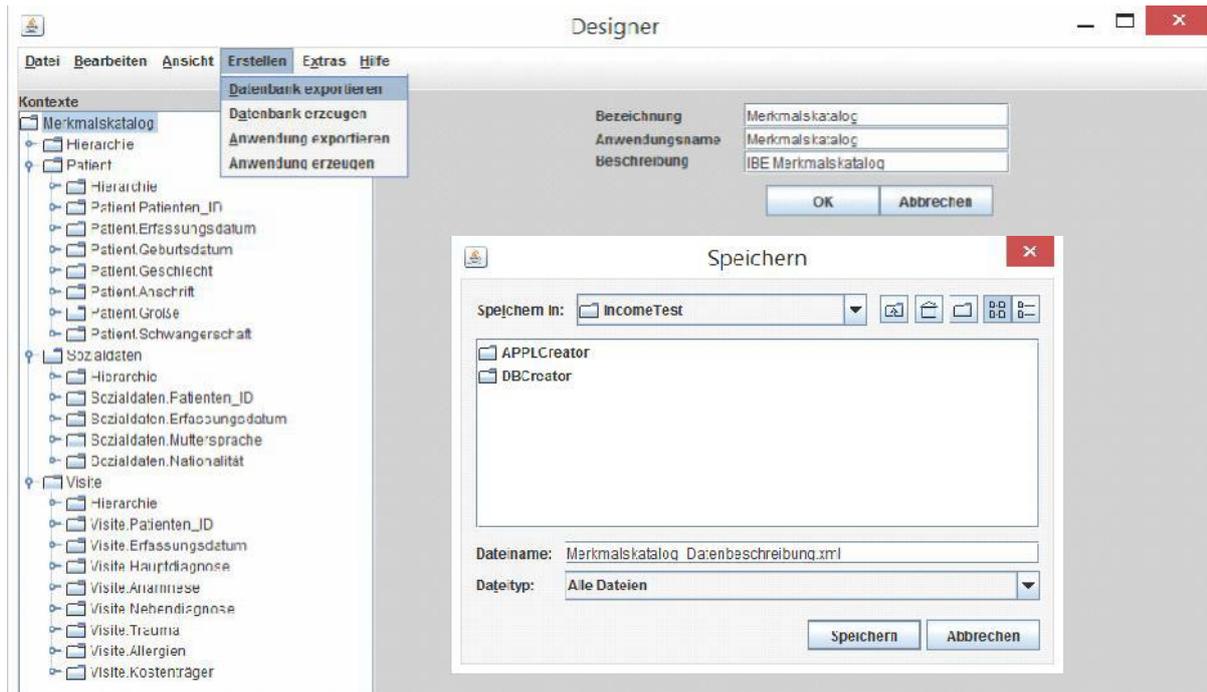
für die Generierung der Datenbeschreibung und eine für die Generierung der Anwendungsbeschreibung) wurden in XSLT geschrieben und von Goertzen zur Verfügung gestellt. Sie werden direkt in das Entwurfsmodul (Designer) eingebunden, um eine Generierung für den Benutzer leicht zu gestalten (s. Abbildung 43). Die Konfiguration des Designers mit diesen Schnittstellen ist Voraussetzung für den Export und die Erzeugung einer MySQL-Datenbank bzw. für den Export und die Erzeugung von den CDMS-Anwendungskonfigurationsdateien. Ein Export für MySQL oder ORACLE ist möglich.



**Abbildung 43: Konfiguration des Designers zur Generierung der Datenbeschreibung und der Anwendungsbeschreibung für das Dokumentationssystem**

#### F.2.8.2 Export der Datenbeschreibung (Datenschema)

Die Generierung der Datenbeschreibung aus dem im Designer modellierten Merkmalskatalog ist unabhängig von einem Datenbankmanagementsystem und somit als XML-Datei zu speichern (s. Abbildung 44). Diese Generierung erfolgt unter dem Menüpunkt Erstellen → Datenbank exportieren. Dies stellt die erste Stufe der Datenbeschreibung dar und kann auch ohne Einbindung der Transformationsschnittstelle generiert werden.



**Abbildung 44: Datenbeschreibung des modellierten Merkmalskatalogs als XML-Datei exportieren**

### F.2.8.3 Export des Datenbankskriptes

Das später aufzusetzende medizinische Dokumentationssystem wird eine Datenbank benötigen, in der die echten Patientendaten gespeichert werden müssen. Die Generierung einer entsprechenden und zu den modellierten Dokumentationsmetadaten passenden Datenstruktur ist deshalb nötig. Die Generierung des Datenskripts erfolgt unter dem Menüpunkt Erstellen → Datenbank erzeugen und wird als SQL-Datei gespeichert (s. Abbildung 45). Dies stellt die zweite Stufe der Datenbeschreibung dar und kann ohne Einbindung der Transformationsschnittstelle „create\_db.xml“ (Datenbank-Plugin) nicht generiert werden. Die im Designer auf Kontext-Ebene modellierten Komponenten (Datenbank, Kontexte, gebundene Merkmalsarten, Stammdaten, Hierarchien und Integritätsbedingungen) werden in die SQL-Sprache übersetzt (s. Tabelle 6). Stammdaten werden zum Beispiel als Datenbank-Tabelle mit sämtlichen modellierten Ausprägungen als Einträge angelegt.

**Tabelle 6: Übersetzung von Designer-Komponenten in SQL-Datenbankbeschreibung**

Designer-Komponente	SQL-Komponente
Database/Anwendung	DATABASE/SCHEMA (Datenbank)
Kontext	TABLE (Tabelle einer Datenbank)
Gebundene Merkmalsart	COLUMN (Spalte einer Tabelle)
Stammdaten	TABLE (Tabellen einer Datenbank (sie enthalten bereits die erfassten Stammdaten))

Designer-Komponente	SQL-Komponente
Hierarchien und Integritätsbedingungen	PRIMARY KEY/FOREIGN KEY (Primär Schlüssel/Fremdschlüssel)

**Beispiel eines SQL-Skripts:**

```
CREATE DATABASE IF NOT EXISTS 'merkmalskatalog'
```

```
USE Merkmalskatalog;
```

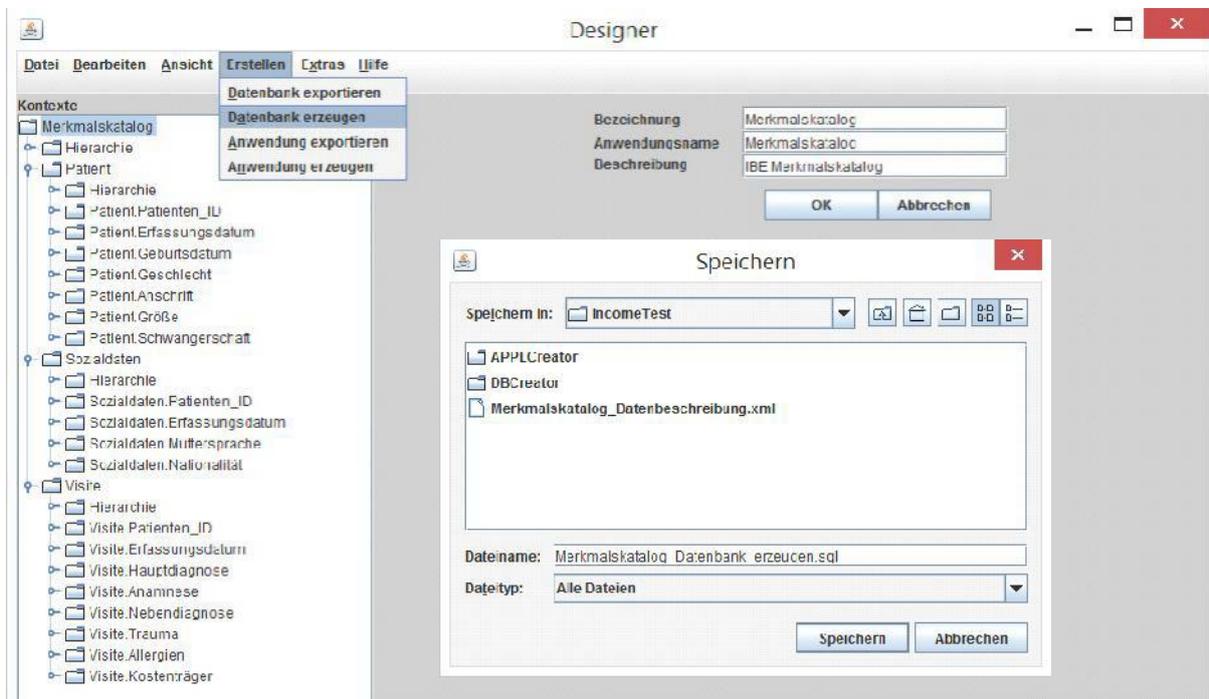
```
CREATE TABLE Patient (Patienten_ID INT NOT NULL, Erfassungsdatum DATE NOT NULL,
Geburtsdatum DATE NOT NULL, Geschlecht INT (10), Anschrift VARCHAR (200), Größe INT (3),
Einheit VARCHAR (20), Schwangerschaft INT (10),
```

```
PRIMARY KEY (Patienten_ID));
```

```
CREATE TABLE Sozialdaten (ID INT NOT NULL, Patienten_ID INT NOT NULL,
Erfassungsdatum DATE NOT NULL, Muttersprache VARCHAR (100), Nationalität INT (10),
PRIMARY KEY (ID));
```

```
/*Erzeugung der Fremdschlüssel auf patient_id aus Tabelle Patient*/
```

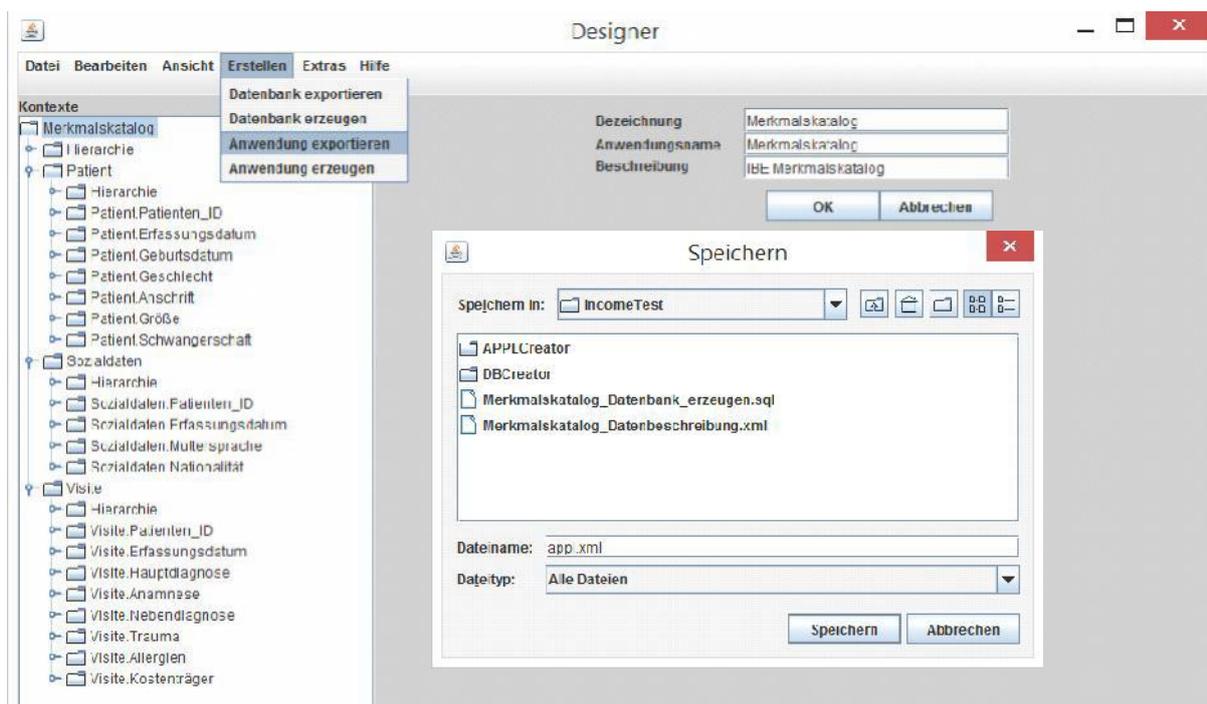
```
ALTER TABLE Sozialdaten ADD CONSTRAINT Patienten_ID_FK FOREIGN KEY(Patienten_ID)
REFERENCES Patient(Patienten_ID);
```



**Abbildung 45: Datenbankskript zur Generierung der CDMS-Datenbank**

#### F.2.8.4 Export der Anwendung (appl.xml)

Auf Grundlage der modellierten Datenbasis wird eine entsprechende vollständige XML-Beschreibung der Dokumentationssystem-Anwendung über dem Menüpunkt Erstellen → Anwendung exportieren erzeugt (s. Abbildung 46). In dieser Datei befinden sich alle im Designer auf Kontext-Ebene modellierten Komponenten (Datenbank, Kontexte, gebundene Merkmalsarten, Stammdaten, Hierarchien und alle Integritätsbedingungen) für die Datenbank Merkmalskatalog. Diese Informationen müssen in einer Datei namens appl.xml gespeichert werden und dienen der Erleichterung der Navigation im späteren Dokumentationssystem. Die appl.xml Datei lässt sich auch ohne Einbindung der Transformationsschnittstellen exportieren.

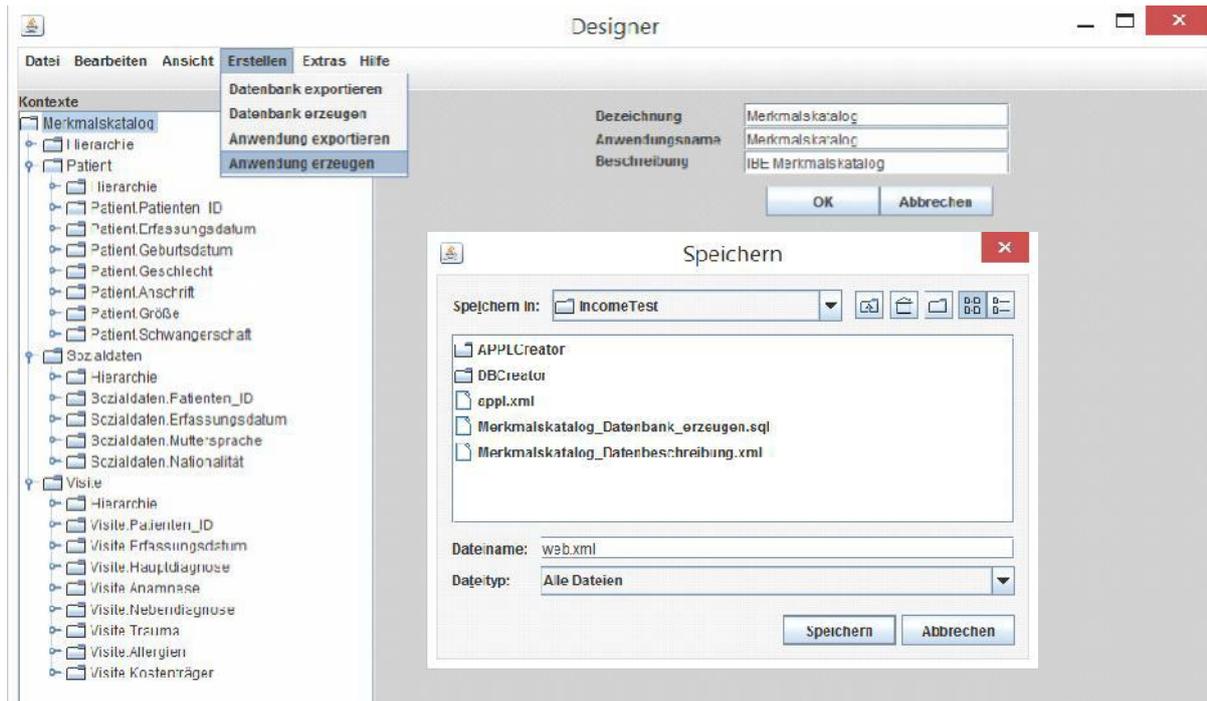


**Abbildung 46: Export der Konfigurationsdatei appl.xml für die CDMS-Dokumentationssystem-Applikation**

#### F.2.8.5 Export der Anwendungskonfiguration (web.xml)

Als letzter Schritt wird die Server-Konfigurationsdatei web.xml zur Festlegung der Funktionalität der Dokumentationssystem-Applikation über den Menüpunkt Erstellen → Anwendung erzeugen generiert (s. Abbildung 47). Beim Start dieser Applikation wird die web.xml automatisch vom Server (der Servlet Container) befragt, um die Applikation im Netzwerk verfügbar zu machen. Die Einbindung der Transformationsschnittstelle „create\_appl.xml“ (Applikations-Plugin) ist für die Generierung der web.xml unerlässlich.

Nach dem Export der Anwendungskonfigurationsdatei wird der Benutzer darauf hingewiesen, dass er diese an sein installiertes Datenbankmanagementsystem anpassen muss (z.B. Treiberangabe, Lokalisation der Datenbank und Datenbankauthentifizierung) (s. Abbildung 48).



**Abbildung 47: Export der Webanwendungskonfigurationsdatei web.xml für die CDMS-Dokumentationssystem-Applikation**



**Abbildung 48: Modifizierungsbedarf der Anwendungskonfigurationsdatei (web.xml)**

### F.2.9 Einschränkungen

1. Bei der Modellierung mit dem VokaDoks-Designer dürfen Bezeichnungen für Merkmalsarten, Stammdaten und Kontexte nicht die Syntax des verwendeten Datenbankmanagementsystems verletzen. Zum Beispiel darf der Begriff „Alter“ nicht benutzt werden, da über den Befehl „ALTER TABLE“ Datenbank-Tabellen in MySQL geändert werden.
2. Die Bezeichnung von Stammdaten sowie der (Feld-)Name von gebundenen Merkmalsarten, Kontexten und Datenbanken dürfen keine Leerzeichen und/oder Umlaute und/oder Sonderzeichen (z.B. ä, ö, ü, ß, /, +, -, \_, etc.) enthalten, um die Datenbank-Grammatik bei der späteren Erzeugung und Nutzung der Dokumentationsdatenbank auf der Anwendungsebene nicht zu verletzen. Der Stammdatenname darf außerdem höchstens 15 Zeichen lang sein.
3. Bei der Erfassung von medizinischen präkoordinierten Datensätzen dürfen jeweils nur zwei Bezeichnungen bzw. Codes aus den angezeigten Konzepten ausgewählt werden.

4. Um ein neues Statement zu erstellen, müssen die vorherigen ausgewählten Begriffe abgewählt und die für die Erstellung des neuen Statements gebrauchten Begriffe ausgewählt werden.
5. Die Entwurf-Arbeit lässt sich lediglich als XML-Dokument mit den modellierten Daten speichern, wenn der ganze Merkmalskatalog erstellt wurde, sprich, wenn die Datenbank (Database) erstellt und mit den Kontexten und Integritätsbedingungen von und zwischen den gebundenen Merkmalsarten fertig modelliert wurde (s. Abschnitt F.2.5).
6. Die Feststellung eines Modellierungsfehlers im CDMS erzwingt eine neue Modellierung mit dem Designer. Deshalb empfiehlt es sich die Entwurf-Arbeit zu speichern und sorgfältig aufzubewahren. Bei der Feststellung eines Modellierungsfehlers im CDMS kann dann diese XML-Datei im Designer wieder geöffnet und Änderungen an der Modellierung vorgenommen werden.

### **F.3 DAS VOKADOKS-BASIERTE DOKUMENTATIONSSYSTEM**

#### ***F.3.1 Einführung***

Die von Goertzen zur Verfügung gestellte ‚leere‘ CDMS-Webanwendung (ohne Patientendaten und ohne klinisch-spezifische Metadaten), das so genannte CDMS-Framework, musste an das VokaDoks-Modell angepasst werden. Die CDMS-Java-Klassen und Eigenschaftsdateien wurden dementsprechend weiterimplementiert und neu kompiliert.

Um ein VokaDoks-basiertes medizinisches Dokumentationssystem (VokaDoks-CDMS) zu installieren und testen, müssen die dafür erforderlichen Dokumentationsmerkmale sowie Integritätsbedingungen mit dem VokaDoks-Designer erfasst, und anschließend Konfigurationsdateien für die Webanwendung eines Dokumentationssystems mit dem CDMS-Creator (Datenbank- und Applikationsplugins) erzeugt werden. Dieser Abschnitt könnte daher als kleines Anwenderhandbuch zum CDMS gesehen werden. Er befasst sich mit dem tatsächlichen und praktischen Einsatz des prototypischen implementierten bzw. entworfenen Dokumentationsmodells.

#### ***F.3.2 Installation eines Anwendungsbeispiels des VokaDoks-medizinischen Dokumentationssystems***

##### **F.3.2.1 Erzeugen der Dokumentationssystem-Datenbank**

Nach der Modellierung von Dokumentationsmerkmalen sowie der Generierung der Webkonfigurationsdateien web.xml und appl.xml und der SQL-Datei zur Generierung der dokumentationsspezifischen Datenbank kann ein VokaDoks-basiertes Dokumentationssystem als Webanwendung installiert werden. Das von Goertzen zur Verfügung gestellte und an das VokaDoks-Modell angepasste Dokumentationssystem, soll zuerst als Webanwendung unter Tomcat installiert werden. Dafür muss es in das Webverzeichnis \webapps\ des Tomcat Installationsverzeichnisses kopiert werden (z.B. unter C:\Programme\Tomcat 7.0\webapps\). Anschließend wird folgendermaßen die Datenbank für das Dokumentationssystem eingerichtet:

### 1. Anlage eines Schemas im verwendeten Datenbankmanagementsystem

Das im Rahmen dieser Arbeit verwendete Datenbankmanagementsystem ist MySQL. MySQL sollte dementsprechend vor jeglichen Unternehmungen als Windows Dienst gestartet werden. Für das entworfene Dokumentationsmodell „Merkmalskatalog“ wird ein gleichnamiges Datenbank-Schema angelegt.

### 2. Import des Datenbankskriptes in die Datenbank

Das im Abschnitt F.2.8.3 generierte Datenbankskript enthält Datenbank-Tabellen und Integritätsbedingungen (bzw. Metadaten) zur Beschreibung und persistenten Speicherung der im Dokumentationssystem eingetragenen (echten Patienten-) Daten. Das Datenbankskript wird in das in MySQL neu angelegte Datenbank-Schema ausgeführt.

### 3. Anlage einer Tabelle zur Benutzerverwaltung

Für die Speicherung von registrierten Nutzern des Dokumentationssystems wird eine neue Tabelle mit dem Namen „user\_admin“ und mit den Feldern „username“ (Name/Login des Benutzers) und „pw“ (Passwort des Benutzers) im Schema Merkmalskatalog angelegt. Somit wird die Authentifizierung von Nutzern auf Anwendungsebene überprüft. Zum Beispiel werden für den Administrator der Benutzername: admin und das Passwort: cdms angelegt.

#### **SQL-Beispiel zur Anlage einer Benutzer-Tabelle:**

```
CREATE TABLE user_admin (username VARCHAR(20) NOT NULL, pw VARCHAR(10) NOT NULL, PRIMARY KEY (username));
INSERT INTO user_admin VALUES ('admin', 'cdms');
GRANT ALL PRIVILEGES ON 'merkmalskatalog'.* TO admin@localhost IDENTIFIED BY 'cdms';
FLUSH PRIVILEGES;
```

#### F.3.2.2 Konfiguration der Dokumentationssystem-Anwendung im Webserver

Folgende Schritte sind für die Konfiguration des Dokumentationssystems als Webanwendung im Webserver zu beachten:

##### 1. Kopieren der Anwendungskonfigurationsdatei web.xml ins Anwendungsverzeichnis

Das im Abschnitt F.2.8.5 generierte web.xml soll in das Webverzeichnis von Tomcat (zum Beispiel unter C:\Programme\Tomcat 7.0\webapps\cdms\WEB-INF\ ) kopiert werden. Vor dem Kopieren soll sichergestellt werden, dass die Syntax der XML-Datei wohlgeformt (aus dem englisch well-formed) ist, d.h. alle XML-Regeln einhält.

##### 2. Eintragung des Datenbanktreibers im web.xml

Folgender Ausschnitt der web.xml Datei zeigt die Eintragung des Datenbanktreibers für eine MySQL-Datenbank, um den Server mit der Datenbank verbinden zu können. Der Speicherort der Datenbank Merkmalskatalog wird ebenfalls eingetragen. Es handelt sich hierbei um die URL, unter welcher die durch die Datenbankerzeugung angelegte MySQL-Datenbank im Netzwerk verfügbar ist. Falls die generierte web.xml eine Standardeinstellung für die Lokalisation der Datenbank beinhaltet, sollte

diese mit aktuellen Einstellungen überschrieben werden. Im web.xml werden außerdem die Logindaten für den Lokalbenutzer eingetragen (s. Abschnitt K.2.4 für ein vollständiges Beispiel der web.xml).

**Ausschnitt aus web.xml:**

```
...
<init-param>
    <param-name>dbDriver</param-name>
    <param-value>com.mysql.jdbc.Driver</param-value>
</init-param>
<init-param>
    <param-name>dbUrl</param-name>
    <param-value>jdbc:mysql://localhost:3306/Merkmalskatalog</param-value>
</init-param>
<init-param>
    <param-name>dbUser</param-name>
    <param-value>admin</param-value>
</init-param>
<init-param>
    <param-name>dbPass</param-name>
    <param-value>cdms</param-value>
</init-param>
...
```

### 3. Speicherung der Applikationskonfiguration appl.xml

Die im Abschnitt F.2.8.4 neu generierte Anwendung appl.xml wird in das installierte CDMS-Framework, z.B. unter C:\Programme\Tomcat 7.0\webapps\cdms\WEB-INF\classes\info kopiert (s. Abschnitt K.2.4 für ein vollständiges Beispiel der appl.xml).

### 4. Bibliotheken

Der Java-Archiv jdom.jar sowie der JDBC-Treiber mysql-connector-java-5.1.22-bin.jar müssen in das Bibliothek-Verzeichnis (\lib) von Tomcat kopiert werden, damit Tomcat die Applikationskonfiguration appl.xml richtig interpretiert und die Verbindung zur Datenbank einwandfrei herstellt.

### 5. Einschränkungen

Die Implementierung der Java-Klassen des von Goertzen zur Verfügung gestellten CDMS-Frameworks wurde nachgebessert bzw. an die Anforderungen der aktualisierten Java-Bibliotheken angepasst. Es wurden ebenfalls neue Java-Klassen hinzugefügt, um der im InCoMe vorgenommenen Erweiterungen zu entsprechen.

### F.3.2.3 Starten und Nutzung des VokaDoks-Dokumentationssystems als Applikation

In diesem Abschnitt wird das generierte medizinische Dokumentationssystem in Form einer in einem Browser lauffähigen integrierten Anwendung gestartet und getestet.

#### 1. Starten des VokaDoks-Dokumentationssystems

Nachdem Tomcat als Windows Dienst gestartet wurde, wird der Tomcat-Manager ebenfalls gestartet und ggf. die Anmelde-Informationen eingegeben. Dies erfolgt zum Beispiel unter Eingabe folgender Adresse in die Adressleiste eines Web-Browsers: `http://127.0.0.1:8282/manager/html`, wobei 8282 die Portnummer ist, unter der Tomcat installiert wurde (standardmäßig ist der Port 8080), und 127.0.0.1 die IP-Adresse des Localhosts (s. Abbildung 49). Anschließend wählt man auf der Tomcat-Manager Seite die VokaDoks-Anwendung, hier „/cdms“ aus. Man gelangt auch direkt zur Anmeldeseite der VokaDoks-Anwendung über die Adresse: `http://127.0.0.1:8282/cdms`.

Nach erfolgreicher Anmeldung (s. Abbildung 50) befindet sich der Benutzer im Kontext „Patient“, der im Designer als höchste Hierarchie der Anwendung definiert wurde. Ein Mausklick auf diesen Kontext ermöglicht dem Benutzer anhand der vordefinierten Suche-Integritätsbedingungen nach Patienten zu suchen (s. Abbildung 51). Ein Klick auf die Suchen-Taste ohne Eingabe von Suchkriterien durchsucht die Dokumentationsdatenbank nach sämtlichen Datensätzen des aktuellen Kontextes (hier Kontext Patient) und zeigt sie an. Abbildung 52 zeigt als Beispiel die Ergebnisse nach einer solchen Suche. Die durch die Ergebnis-Integritätsbedingungen definierten Merkmalsarten für den Kontext Patient werden angezeigt.

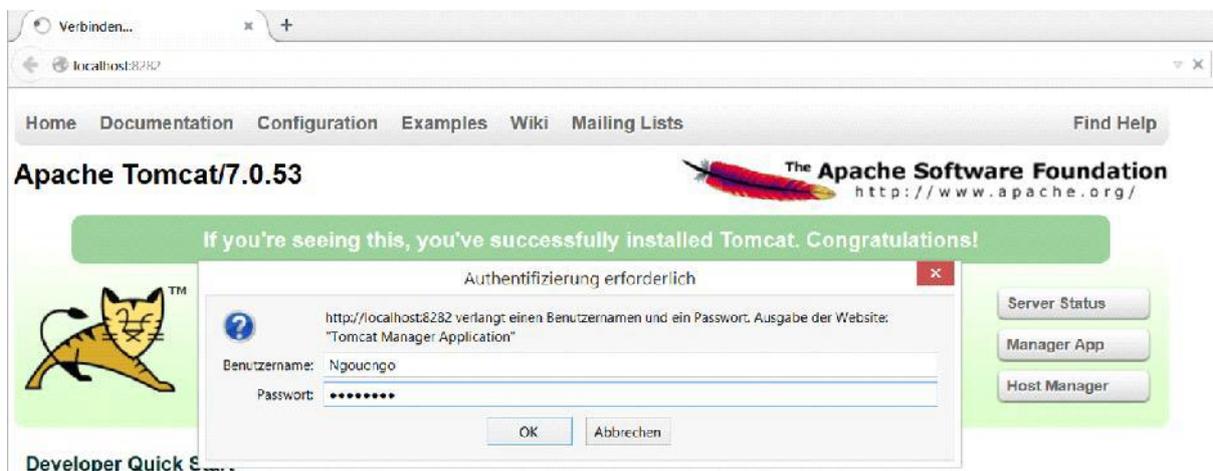


Abbildung 49: Anmeldung beim Tomcat-Manager

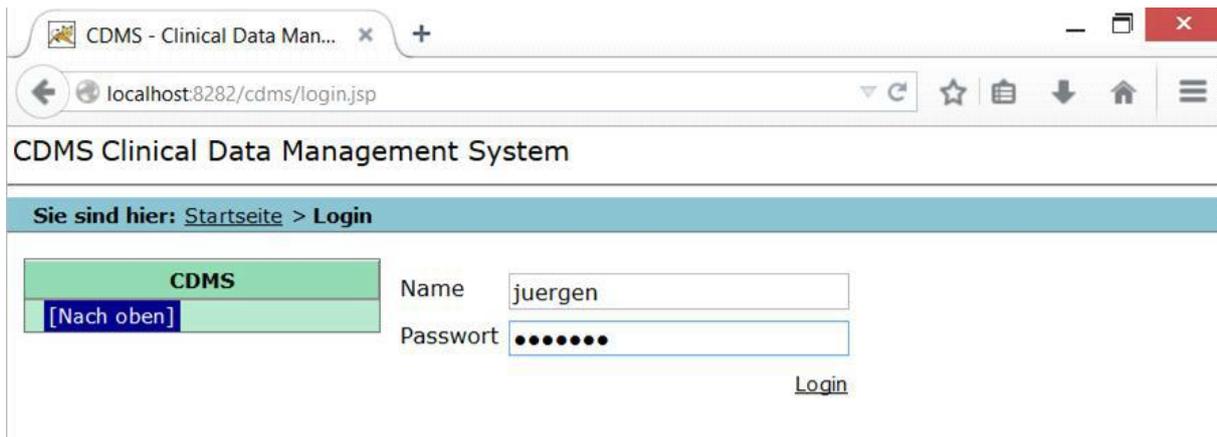


Abbildung 50: CDMS Anmeldung

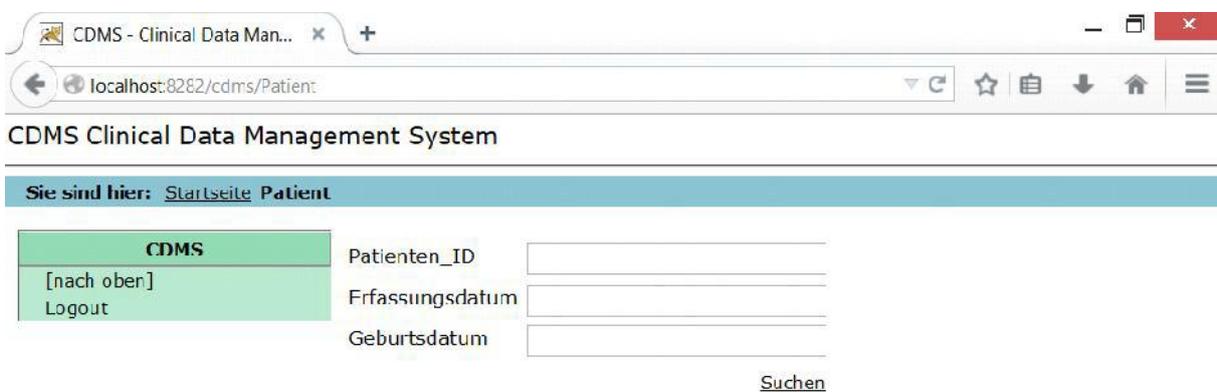
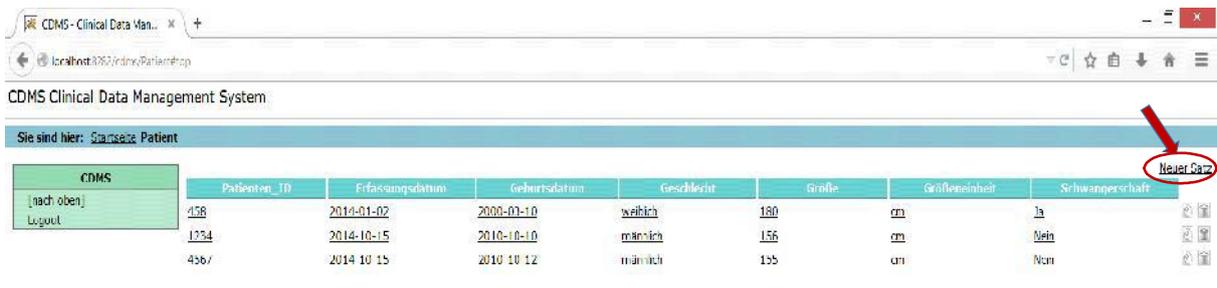


Abbildung 51: Patientensuche nach den vordefinierten Suchkriterien

2. Anlegen eines neuen Patienten

Falls die erste Suche nach Patienten eine leere Menge oder eine Liste in der Datenbank vorhandener Patienten ergibt, hat der Benutzer immer die Möglichkeit einen neuen Patienten anzulegen (s. Abbildung 52 und Abbildung 53). Stammdaten werden als Auswahlliste (Drop-Down Liste) angezeigt. Der erste Wert (d.h. Wert in der ersten Position in der Liste) wird als Default Wert der Liste angezeigt. Als Eingabehilfe werden die richtigen Messeinheiten direkt angezeigt (keine Auswahlliste!). Bei Änderungen dieser wird eine Fehlermeldung angezeigt.



### Abbildung 52: Ergebnisliste nach einer Suche nach Patienten und Möglichkeit einen neuen Patienten anzulegen

The screenshot shows a web browser window with the address bar displaying 'localhost:6282/cdms/Patient'. The page title is 'CDMS Clinical Data Management System'. Below the title, there is a breadcrumb trail: 'Sie sind hier: Startseite Patient'. On the left side, there is a green navigation menu with the following items: 'CDMS', '[nach oben]', and 'Logout'. The main content area contains a form for creating a new patient with the following fields:

Patienten_ID	<input type="text"/>
Erfassungsdatum	<input type="text"/>
Geburtsdatum	<input type="text"/>
Geschlecht	männlich <input type="button" value="v"/>
Anschrift	<input type="text"/>
Größe	<input type="text"/>
Größeneinheit	cm <input type="text"/>
Schwangerschaft	Ja <input type="button" value="v"/>

At the bottom of the form, there are two buttons: 'Zurück' and 'Speichern'.

### Abbildung 53: Form zum Anlegen eines neuen Patienten

#### 3. Detailanzeige eines Patienten

Falls man die Daten eines bestimmten Patienten näher betrachten möchte, wählt man den Patienten aus der Ergebnisliste durch einen einfachen Klick darauf aus (s. Abbildung 54). Es bestehen hier keine Änderungsmöglichkeiten. Über den Button „zurück“ oder „[nach oben]“ in der Navigationsleiste gelangt man zurück in die Ergebnisliste des jeweiligen Kontexts.

Der Kontext Patient wurde als höchste Hierarchie der Anwendung modelliert. Kontexte der unteren Hierarchieebene werden nur in der Detailanzeige eines ausgewählten Patienten angezeigt. Zu einem Patienten können nämlich noch seine Visiten und seine Sozialdaten erzeugt werden. Zu einem Patienten können außerdem mehr als eine Visite sowie mehrere Sozialdaten erfasst werden (diese hängt allerdings von der semantischen Modellierung im Designer ab). Abbildung 55 und Abbildung 56 zeigen jeweils Visiten- und Sozialdaten-Datensätze eines ausgewählten Patienten.



Abbildung 54: Detailanzeige eines Patientendatensatzes



Abbildung 55: Visiten eines Patienten

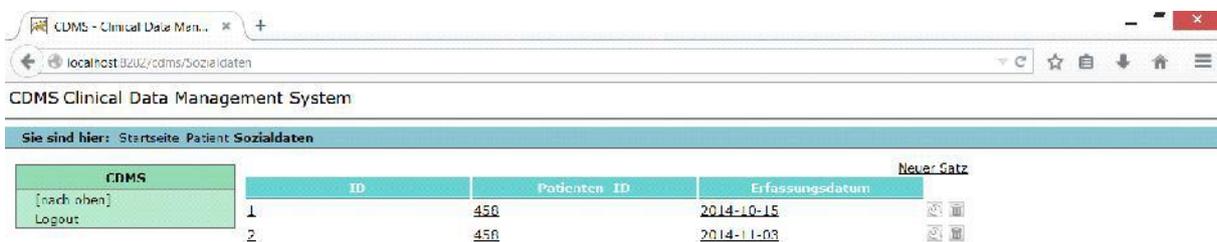


Abbildung 56: Sozialdaten eines Patienten

#### 4. Bearbeiten eines Patienten

Möchte man die Daten eines Patienten bearbeiten, klickt man auf das „Gabelschlüssel“-Symbol neben dem zu bearbeitenden Patienten. So gelangt man zu einer Detailanzeige des Datensatzes mit Änderungsmöglichkeiten (s. Abbildung 57).

CDMS Clinical Data Management System

Sie sind hier: [Startseite](#) [Patient](#)

<b>CDMS</b>	Patienten ID	158
[nach oben]	Erfassungsdatum	2014-01-02
Logout	Geburtsdatum	2000-03-10
	Geschlecht	weiblich
	Anschrift	Rosenstr. 12, 69133 Mannheim
	Größe	180
	Größeneinheit	cm
	Schwangerschaft	Ja

[Zurück](#) [Speichern](#)

**Abbildung 57: Form mit Änderungsmöglichkeit eines Patientendatensatzes**

#### 5. Löschen eines Patienten

Möchte man die Daten eines Patienten löschen, klickt man auf das „Mülleimer“-Symbol neben dem zu löschenden Patienten. So gelangt man zu einer Detailanzeige mit der Möglichkeit den Patientendatensatz zu löschen (s. Abbildung 58).

CDMS Clinical Data Management System

Sie sind hier: [Startseite](#) [Patient](#)

<b>CDMS</b>	Patienten_ID	4567
[nach oben]	Erfassungsdatum	2014-10-15
Logout	Geburtsdatum	2010-10-12
	Geschlecht	männlich
	Anschrift	rosenstr. 15
	Größe	155
	Größeneinheit	cm
	Schwangerschaft	Nein

[Zurück](#) [Löschen](#)

**Abbildung 58: Form zum Löschen eines Patientendatensatzes**

#### 6. Anlegen von Kontexten der unteren Hierarchieebene

Wählt man aus dem Patientenmenü (Menü des Kontextes in der höchsten Hierarchie über die Navigationsleiste auf der linken Seite) einen Subkontext aus, bekommt man die Möglichkeit einen neuen Datensatz für diesen Subkontext anzulegen. Abbildung 59 und Abbildung 60 zeigen als Beispiel die Anlage einer neuen Visite bzw. eines neuen Sozialdaten-Datensatzes für den Patienten mit der Patientennummer 458.

Sie sind hier: Startseite Patient Visite

<b>CDMS</b>	ID	<input type="text"/>
[nach oben]	Patienten_ID	458
Logout	Erfassungsdatum	<input type="text"/>
	Hauptdiagnose	Keuchhusten
	Anamnese	<input type="text"/>
	Nebendiagnose	Rückfallfieber Fleckfieber
	Hat der Patient eine Keuchschüttelung?	Ja
	Hat der Patient Allergien?	Ja
	Kostenträger	AOK

[Zurück](#) [Speichern](#)

**Abbildung 59: Form zum Anlegen einer neuen Visite zu einem bestimmten Patienten**

Sie sind hier: Startseite Patient Sozialdaten

<b>CDMS</b>	ID	<input type="text"/>
[nach oben]	Patienten ID	458
Logout	Erfassungsdatum	<input type="text"/>
	Muttersprache	<input type="text"/>
	Nationalität	Kamerun

[Zurück](#) [Speichern](#)

**Abbildung 60: Form zum Anlegen von neuen Sozialdaten eines bestimmten Patienten**

## 7. Syntaktische und semantische Fehlermeldungen

Beim Anlegen sowie beim Bearbeiten eines Kontext-Datensatzes wird stets die Dateneingabe nach den hinterlegten Integritätsbedingungen überprüft. Bei Verletzung dieser Integritätsbedingungen werden die entsprechenden modellierten Fehlermeldungen angezeigt. Abbildung 61 und Abbildung 62 zeigen als Beispiel syntaktische und semantische Fehlermeldungen bei Fehlangaben während des Anlegens eines Patienten, so wie sie im Designer modelliert wurden. Bei der syntaktischen Fehlermeldung in Abbildung 61 geht es um die Formatierung des Datums. Bei der semantischen Fehlermeldung in Abbildung 62 muss das Erfassungsdatum nach dem Geburtsdatum eines Patienten liegen. Männliche Patienten dürfen nicht die Ausprägung „Schwangerschaft: Ja“ haben. Außerdem wurde die Messeinheit für die Körpergröße fälschlicherweise geändert. Kein Datensatz darf gespeichert werden, ohne vorherige Behebung der dargestellten Fehler. Warnmeldungen stellen jedoch kein Hindernis zur Speicherung der Daten dar.

CDMS - Clinical Data Man... x +

localhost:8282/cdms/Patient

CDMS Clinical Data Management System

Sie sind hier: Startseite Patient

<b>CDMS</b>	Patienten_ID	8524	
[nach oben]	Erfassungsdatum	20.11.2014	Fehler in Feld Erfassungsdatum: Falscher Datentyp. Datentyp ist Datum im Format JJJJ-MM-TT.
Logout	Geburtsdatum	05.12.2002	Fehler in Feld Geburtsdatum: Falscher Datentyp. Datentyp ist Datum im Format JJJJ-MM-TT.
	Geschlecht	männlich	
	Anschrift	Rosenstr. 90, 81377 München	
	Größe	180	
	Größeneinheit	cm	
	Schwangerschaft	Nein	
	<a href="#">Zurück</a>		<a href="#">Speichern</a>
	<a href="#">Bitte Meldungen beachten</a>		

Abbildung 61: Beispiel einer Fehlermeldung bei Verletzung einer syntaktischen Integritätsbedingung: Das Datumformat für das CDMS ist JJJJ-MM-TT.

CDMS - Clinical Data Man... x +

localhost:8282/cdms/Patient

CDMS Clinical Data Management System

Sie sind hier: Startseite Patient

<b>CDMS</b>	Patienten_ID	8524	
[nach oben]	Erfassungsdatum	18.11.2014	Fehler in Feld Erfassungsdatum: Erfassungsdatum muss größer als Geburtsdatum sein. Datenfeld muss größer 20.12.2014 sein.
Logout	Geburtsdatum	20.12.2014	Fehler in Feld Geburtsdatum: Erfassungsdatum muss größer als Geburtsdatum sein. Datenfeld muss kleiner 18.11.2014 sein.
	Geschlecht	männlich	Fehler in Feld Geschlecht: Geschlecht muss bei Schwangerschaft weiblich sein. Datenfeld muss 2 sein (also 2. Position in der Auswahlliste).
	Anschrift	Rosenstr. 90, 81377 München	
	Größe	180	
	Größeneinheit	m	Fehler in Feld Einheit: Messeinheit ungültig. Datenfeld Wert ist: cm
	Schwangerschaft	Ja	
	<a href="#">Zurück</a>		<a href="#">Speichern</a>
	<a href="#">Bitte Meldungen beachten</a>		

Abbildung 62: Beispiel einer Fehlermeldung bei Verletzung einer semantischen Integritätsbedingung

## G DISKUSSION

---

Angesichts der im Rahmen dieser Arbeit festgestellten Problematik bezüglich des Mankos von Dokumentationssystemen mit standardisierten kontrollierten Vokabularen, lag die Motivation dieser Arbeit in der Leistung eines Beitrags in der medizinischen Forschung und Patientenversorgung und in der Darbietung nützlicher Anregungen, um die Patientendatenqualität und somit die Patientenpflege zu verbessern. Ziel der vorliegenden Arbeit war also die Entwicklung eines im Gesundheitswesen benötigten integrierten Dokumentationsmodells zur Bereitstellung von Semantiken für Dokumentationseinheiten, zur Vereinheitlichung von Daten-Wertebereichen, und zur Standardisierung von Dokumentationsmerkmalsarten durch den Einsatz von standardisierten kontrollierten Vokabularen. Somit ist das Dokumentationsmodell VokaDoks entstanden, das die Entwicklung eines terminologisch fundierten Dokumentationssystems erleichtern soll. Die Anwendbarkeit und der Nutzen dieses Modells wurden schließlich durch eine Implementierung und eine prototypische Anwendung belegt.

Recherchen in der aktuellen Literatur zeigen, dass Standardisierungsbemühungen von Dokumentationssystemen weiterhin zunehmen. Andere Bemühungen betreffen die Suche nach einer vereinfachten aber präzisen, effizienten und vollständigen standardisierten Terminologie/Klassifikation zur bestmöglichen Abdeckung der klinischen Dokumentation [3, 16, 40, 46, 70]. Diese steigende Tendenz bestätigt zum einen die Relevanz des Themas dieser Arbeit und zum anderen dessen Aktualität. Die angewandten Ansätze betreffen oft die Analyse der Verwendbarkeit von ausgewählten Terminologien oder Klassifikationssystemen [3, 12, 16] durch Mapping von Dokumentationseinheiten und umgekehrt. In seinen Bemühungen analysierte zum Beispiel Feng die Abbildbarkeit von Pflegedokumentationsdatensätzen in dem Clinical Care Classification (CCC) System in einem medizinischen Zentrum in Taiwan [16], um einerseits die Eignung, die Nutzung und den Nutzen dieses klinischen Informationssystems für die Pflegedokumentation zu zeigen und andererseits, um implizit die Notwendigkeit der Standardisierung von (Pflege-) Dokumentationssystemen zu zeigen. Der im Rahmen dieser Arbeit angewandte Top-Down-Ansatz und Bottom-Up-Ansatz verfolgte sowohl das Ziel das Vermögen durch kontrollierte Vokabulare klinische Dokumentationsmerkmale abzudecken als auch ein Kernmodell für die Entwicklung von terminologisch fundierten medizinischen Dokumentationssystemen zu entwickeln. Besonders bei dem Entwurf dieses Dokumentationsmodells war die Umsetzung im Modell der Modellierungsstärken von den folgenden fünf analysierten Ansätzen:

- Das ISO/IEC 11179 Ed. 3 Metamodell für seine syntaktische und semantische Beschreibungsvermögen von Dokumentationsmerkmalen, sowie für die Bereitstellung von Modellen zur Einbindung von kontrollierten Vokabularen,
- TERMTrial für seine anziehende Modellierungsart, was die Trennung von allgemeinen und studienspezifischen Begriffen im Terminologie-Managementsystem anbelangt,

- ORCA für seine besondere Art Dokumentationsmerkmale durch den Aufbau einer Wissensbasis mit einem medizinischen Thesaurus zu spezifizieren,
- Das Seoulsche Terminologie-basierte ENRS für seine spezifische Art Dokumentationsmerkmale durch sogenannte „precoordinated statements“ zu definieren,
- InCoMe für seine ganz spezielle Art allgemeine Dokumentationsmerkmale und Stammdaten für Dokumentationsmerkmale zu definieren, um sie später je nach Anwendungskontext zu benutzen. Ganz speziell bei InCoMe ist ebenfalls die Berücksichtigung von Integritätsbedingungen, dessen Nutzen in der vorliegenden Arbeit verwertet wurde.

Bei der Modellierung des VokaDoks wurden die gleichen Analyse Kriterien und Anforderungen berücksichtigt, die bei der Analyse bestehender Ansätze zur terminologischen Fundierung von Dokumentationssystemen angewandt wurden, nämlich die Nutzung eines kontrollierten Vokabulars, die Definition einer Merkmalsebene, die Definition von Wertebereichen, und die Definition von Rollen und Entitäten. Tabelle 7 fasst deshalb zum Vergleich und aus Integritätsgründen alle Ansätze einschließlich VokaDoks nach den Anforderungen an ein terminologisch fundiertes Dokumentationssystem zusammen.

**Tabelle 7: Vergleich von VokaDoks mit den in der vorliegenden Arbeit analysierten Ansätzen anhand der Analyse Kriterien.**

	Vokabular	Rollen	Merkmalswelt	Speicher für die Wertebereiche
ISO 11179 Ed.3	✓	✓	?	✓
TERMTrial	?	✓	✓	✓
InCoMe	X	✓	✓	✓
ORCA	?	✓	✓	X
ENRS	?	✓	✓	?
VokaDoks	✓	✓	✓	✓

Zur Erfüllung der Analyse Kriterien besteht das VokaDoks-Dokumentationsmodell aus drei Hauptstrukturen:

- Das *Vocabularies System* zur Verwaltung von kontrollierten Vokabularen, sowie von nicht standardisierten medizinischen Begriffen,
- Das *Clinical Documentation System* zur Modellierung von Dokumentationsmerkmalen und Wertebereichen für Dokumentationsmerkmale, sowie zur Festlegung der Rolle einzelner Dokumentationsmerkmale je nach modelliertem medizinischen Kontext.

- Die *Vocabularies-based Documentation Database* zur Pflege der modellierten Dokumentationsmetadaten und Ausprägungen aus Konzepten des *Vocabularies Systems* sowie zur persistenten Speicherung der echten Dokumentationsdaten (wie z.B. die erhobenen Patientendaten).

Der Beleg der Anwendbarkeit des VokaDoks-Dokumentationsmodells durch die Implementierung einer prototypischen Anwendung zeigte jedoch Einschränkungen. Diese Implementierung lehnte sich an den zur Verfügung gestellte InCoMe-Designer sowie an das InCoMe-CDMS, welche zur Erfüllung der Analyse Kriterien weiter ausgebaut wurden. So wurde zum Beispiel die Einbindung des Moduls *Clinical Thesaurus* des VokaDoks-Dokumentationsmodells in den Ausbau des InCoMe-Designers nicht umgesetzt. Der Schwerpunkt bei diesem Ausbau wurde auf die Umsetzung der Hauptmodule *Vocabularies System* und *Clinical Documentation System* des VokaDoks gelegt, da sie die Kernkomponenten eines terminologisch fundierten Dokumentationssystems darstellen und für die Verwaltung von kontrollierten Vokabularen, sowie die Modellierung von Dokumentationsmerkmalen und von Wertebereichen für Dokumentationsmerkmale essenziell sind.

Außerdem bringt die Trennung von der Modellierung von Dokumentationsmerkmalen in der eigenen Entwurf-Applikation (Designer) und der konkreten Dokumentationssystem-Anwendung (CDMS) viele Vorteile mit sich. Durch die Integration der im Designer erzeugten Konfigurationskomponenten in das von Goertzen zur Verfügung gestellte "leere" CDMS gibt es nämlich folgende Vorteile:

- Keine Notwendigkeit, die erstellte Applikation neu zu kompilieren,
- Die Dokumentationssystem-Anwendung ist Plattform-unabhängig und somit auf unterschiedlichsten Betriebssystemen lauffähig,
- Die Dokumentationssystem-Anwendung ist nach Installation unter Berücksichtigung der Anforderungen der angewandten Software (z.B. Webcontainer Tomcat) sofort lauffähig,
- Der Generierungsaufwand ist durch die Wiederverwendung von bereits modellierten Dokumentationsmerkmalen gering (s. F.2.7). Somit erhöhen sich die Wartbarkeit und die Stabilität der Anwendung.

Ein weiterer Vorteil besteht in dem Grad an Freiheit, den die prototypische Implementierung des VokaDoks-Dokumentationsmodells bietet. Es bleibt immer die Möglichkeit, bei Bedarf, andere Programmiersprachen bzw. andere Datenbankmanagementsysteme zu nutzen, indem die im Designer einzubettenden Transformationsskripte zur Generierung der Daten- und Applikationsbeschreibung entsprechend angepasst oder nachgebessert werden. Diese Flexibilität wird vor allem durch den Einsatz von XML als Beschreibungsmittel der Modellierung geboten. Somit sind die Erweiterbarkeit der implementierten VokaDoks-Komponenten sowie eine angenehme Datenmanipulation sichergestellt. Die semantische Interoperabilität zwischen unterschiedlichen Dokumentationssystemen wird ebenfalls dadurch unterstützt.

Die in der Wissenschaft empfohlene Wiederverwendung von Daten bzw. Metadaten wird ebenfalls im *Clinical Documentation System* des VokaDoks durch die Modellierung von sogenannten freien

Merkmalsarten unterstützt [26, 27, 40, 47]. Diese allgemeinen Merkmalsarten können nämlich beliebig oft für die Erstellung von medizinischen Kontexten benutzt werden.

Die Verwendung von Cascading Style Sheets (CSS) zur Gestaltung von HTML-Seiten und von Java Server Pages (JSP) zur dynamischen Erzeugung von HTML- und XML-Ausgaben bei der Implementierung der CDMS-Webanwendung bietet außerdem die Möglichkeit das Design des generierten Dokumentationssystems an die Bedürfnisse des Endbenutzers anzupassen. Dafür sind jedoch Vorkenntnisse in diesen Programmiersprachen erforderlich. Die graphische Benutzeroberfläche des von Goertzen zur Verfügung gestellten Designers ist hingegen statisch konzipiert, da sie nicht für den Endbenutzer gedacht war.

Es ist jedoch nach Cornet et al. schwierig alle Strukturen bzw. Anforderungen eines klinischen Anwendungsgebiets in einem Modell zu berücksichtigen, da sie unter Umständen oft kompliziert und vor allem umfangreich sind [13]. Und angesichts der stets wachsenden Zahl von Anforderungen an die medizinische Dokumentation [40] konnte das in der vorliegenden Arbeit vorgestellte Dokumentationsmodell nicht sämtliche speziellen klinischen Anforderungen berücksichtigen. Dafür würde das VokaDoks-Dokumentationsmodell zusätzliche Erweiterungen in der Implementierung sowie zusätzliche Tests benötigen. Aus dem neuesten Stand der Literatur ist es ersichtlich, dass Bemühungen Datenbestände der Medizin sowie verschiedene medizinische kontrollierte Vokabulare zu standardisieren und vor allem zu harmonisieren die Qualität von medizinischen Dokumentationssystemen deutlich erhöhen [40, 46, 70]. Die Harmonisierung des *Vocabularies Systems* des VokaDoks-Modells könnte dementsprechend in Folgearbeiten erzielt werden. Eine Anreicherung des *Vocabularies Systems* mit weiteren standardisierten medizinischen bzw. klinischen Terminologien und Klassifikationen würde dabei das Potenzial des VokaDoks-basierten CDMS erhöhen. Die im Rahmen dieser Arbeit zur Verfügung gestellten Funktionalitäten des VokaDoks-basierten CDMS reichen jedoch aus, um ein standardisiertes allgemeines medizinisches Dokumentationssystem, welches auf Terminologien basiert, zu gestalten und es effektiv zu nutzen.

Nach dem vom Rosenbloom definierten Modell zur Evaluierung von Interface-Terminologien [61] und gemäß den von Bakhshi-Raiez aufgeführten fünf Aspekten der Nutzbarkeit eines Systems – Effektivität, Effizienz, Erlernbarkeit, Nutzerzufriedenheit und aufgetretene Probleme [3] – sollte das entwickelte VokaDoks-Dokumentationsmodell der Bewertung von Dritten unterliegen. Eine mögliche Evaluierungsfrage wäre zum Beispiel zu wissen, ob das VokaDoks-Dokumentationsmodell alle von Rosenbloom vorgestellten Terminologie-Attribute abbildet. Eine tiefe Analyse sollte in dieser Hinsicht von Endbenutzern durchgeführt werden, um die Verwendbarkeit bzw. Nutzbarkeit der VokaDoks-Web-Anwendung für ihr spezielles medizinisches Gebiet gemäß ihren Rahmenbedingungen zu prüfen.

## H AUSBLICK

---

Die im Rahmen dieser Arbeit vorgestellten Konzepte zur Entwicklung eines terminologisch fundierten Dokumentationssystems haben bestätigt, wie unerlässlich die Einbettung kontrollierter Vokabulare in medizinische Dokumentationssysteme sind, um die Effizienz, die Nutzbarkeit-Faktoren von Dokumentationssystemen und somit die Benutzerzufriedenheit zu verbessern. Dabei verbessert sich ebenfalls die Qualität der erfassten bzw. verwalteten Daten. Dokumentationssysteme müssen nämlich exakte und vollständige Daten bereitstellen. Diese Arbeit hat ebenfalls gezeigt, dass eine durchgängige und übergreifende Sicherung der Datenqualität und der Qualität eines medizinischen Dokumentationssystems durch eine computerunterstützte Generierung von Dokumentationssystemen möglich ist. Die im Rahmen dieser Arbeit festgestellten Einschränkungen lassen sich auf die Einschränkungen durch die verwendeten Software-Systeme zurückführen. So war zum Beispiel eine vollständige maschinelle Umsetzung bei der Generierung der Anwendungs- und Konfigurationsdateien nicht möglich [20]. Der Benutzer wurde zum Beispiel an dieser Stelle benachrichtigt, die Dateien manuell anzupassen (z.B. Eintragung des verwendeten Datenbanktreibers im web.xml). Leichte Vorkenntnisse in der Datenbank-Welt werden außerdem vorausgesetzt, um das erzeugte Datenbank-Skript in ein Datenbankverwaltungssystem auszuführen.

Dank dem Ausbau des von Goertzen zur Verfügung gestellten Entwurf-Moduls und CDMS-Frameworks gemäß den Konzepten und dem Modell, das im Rahmen dieser Arbeit entwickelt wurde, konnten tatsächlich folgende Erweiterungen implementiert werden:

- Ergänzung zusätzlicher Formen von Integritätsbedingungen, wie die Einheitsintegritätsbedingung,
- Implementierung von Schnittstellen zur Nutzung kontrollierter Vokabulare,
- Ergänzung von Schnittstellen zur Restrukturierung der Dokumentationssystem-Datenbank,
- Einbettung neuer Datenfelder im Designer.

Die Implementierungsart des Entwurf-Moduls und des CDMS-Frameworks bietet jedoch mehr Erweiterungsmöglichkeiten an. Beide Module besitzen viel Potenzial, das im Rahmen von Folgeprojekten ausreichend ausgenutzt werden kann, um ein durchaus einzigartiges und an zukünftige medizinische Anforderungen angepasstes medizinisches Dokumentationssystem zu erzeugen. Folgende Erweiterungsmöglichkeiten, wie sie bereits von Goertzen angekündigt wurden [20], sind durch diese Arbeit bestätigt:

- Import und Export von Stammdaten und freien Merkmalsarten (dies würde die Wiederverwendung von Merkmalen und Stammdaten sowie die Austauschmöglichkeit (semantische Interoperabilität) zwischen unterschiedlichen Dokumentationssystemen deutlich erhöhen),
- Import bzw. Implementierung von Schnittstellen zu anderen Dokumentationsumgebungen,
- Einbeziehung von Formeln in die Definition von semantischen Integritätsbedingungen.

Zu guter Letzt bietet das im Rahmen dieser Arbeit vorgestellte VokaDoks-Dokumentationsmodell Module, wie das *Clinical Thesaurus*, deren Implementierung durchaus vorteilhaft wäre.

## I ZUSAMMENFASSUNG

---

Die medizinische Dokumentation spielt eine zentrale Rolle in der Patientenversorgung und einer immer wachsenden vernetzten medizinische und klinische Forschung. Die im Rahmen dieser Arbeit verfolgten Ziele waren deshalb ein Modell zur Integration von Datenhaltung, Dokumentation und Terminologie in einem medizinischen Dokumentationssystem zu erarbeiten und die Anwendbarkeit des integrierten Dokumentationsmodells durch eine Implementierung und prototypische Anwendung zu belegen. Die Relevanz und Stärke sowie der Einsatz von standardisierten medizinischen kontrollierten Vokabularen in unterschiedlichen Dokumentationssystemen wurde somit anhand zweier Ansätze, Top-Down and Bottom-Up, sowie anhand der Modellierung eines terminologisch fundierten Dokumentationssystems namens VokaDoks überprüft und bestätigt. Standardisierte medizinische kontrollierte Vokabulare kommen den Anforderungen an einem Dokumentationssystem entgegen. Ein medizinisches Dokumentationssystem soll nämlich relevante Patienten- bzw. medizinische Daten so präzise und zuverlässig wie möglich darstellen und zur Verfügung stellen, um die Patientenversorgung sowie die Durchführung von Forschungsvorhaben zu verbessern. Es wurde außerdem gezeigt, dass standardisierte medizinische kontrollierte Vokabulare die Qualität der zu erhebenden Daten erhöhen.

In der prototypischen Implementierung eines terminologisch fundierten medizinischen Dokumentationssystems, das auf das vorgestellte Dokumentationsmodell VokaDoks basiert, wurde versucht die drei strukturellen Hauptkomponenten des VokaDoks in den von Goertzen zur Verfügung gestellten CDMS-Designer zu integrieren. Eine Vorarbeit bestand darin, VokaDoks-Modellkomponenten durch CDMS-Komponenten des Goertzen-Dokumentationsmodells InCoMe abzubilden. Ein erfolgreiches Mapping ermöglichte den Ausbau von InCoMe mit den Komponenten

- *Vocabularies System* zur Verwaltung von kontrollierten Vokabularen,
- *Clinical Documentation System* (VokaDoks-Designer) zur Modellierung von Dokumentationsmerkmalen und Wertebereichen für Dokumentationsmerkmale, sowie zur Festlegung der Rolle einzelner Dokumentationsmerkmale je nach modelliertem medizinischen Kontext,
- *Vocabularies-based Documentation Database* zur Pflege der modellierten Dokumentationsmetadaten sowie zur persistenten Speicherung der echten Daten. Alle aus dem Designer zu exportierenden Skripte befinden sich im Anhang K.2.

Das *Vocabularies System* wurde mit den kontrollierten Vokabularen ICD-10-GM 2010, OPS 2010, TNM 2006, MedDRA v.13.0 und SNOMED-CT 2010 bereichert und in das *Clinical Dokumentation System* eingebunden. Im *Clinical Dokumentation System* konnten Merkmalsarten sowie Stammdaten unter Berücksichtigung von syntaktischen und semantischen Integritätsbedingungen erfasst und miteinander verknüpft werden. Zum Schluss konnten nach Bündelung von Kontexten eines spezifischen medizinischen Gebiets Konfigurationsdateien exportiert werden, die zum Aufbau des konkreten VokaDoks- und webbasierten Dokumentationssystem dienen werden. Der Export einer speziellen Konfigurationsdatei, nämlich des Datenbankskriptes diente dazu die *Vocabularies-based*

*Documentation Database* in einem MySQL oder ORACLE Datenbankmanagementsystem zu erzeugen. Hierbei bestehen Erweiterungsmöglichkeiten der angewandten XSLT-Transformationsschnittstellen, um die *Vocabularies-based Documentation Database* in anderen Datenbankmanagementsystemen wie PostGreSQL zu erzeugen.

Es gab allerdings Einschränkungen bei der Modellierung, die jedoch die Anwendbarkeit des Modells nicht einschränken, sondern einerseits zu der optimalen Nutzung des VokaDoks-Designers dienen und andererseits als Hinweis für den Datenmanager bzw. den Endbenutzer gelten.

**J LITERATURVERZEICHNIS**

---

- [1] L. Ahmadian, M. van Engen-Verheul, F. Bakhshi-Raiez, N. Peek, R. Cornet, N.F. de Keizer, The role of standardized data and terminological systems in computerized clinical decision support systems: literature review and survey, *Int J Med Inform* 80 (2011) 81-93.
- [2] Arzneimittelkommission der deutschen Ärzteschaft, Hrsg. Pharmakovigilanz. Arzneiverordnung in der Praxis, Band 32, Sonderheft 1, April 2005. Berlin: Arzneimittelkommission der deutschen Ärzteschaft, 2005.
- [3] F. Bakhshi-Raiez, N.F. de Keizer, R. Cornet, M. Dorrepaal, D. Dongelmans, MW. Jaspers, A usability evaluation of a SNOMED CT based compositional interface terminology for intensive care, *Int J Med Inform* 81(5) (2012) 351-362.
- [4] C. Ball, Beyond Data about data: The litigator's guide to metadata, at 2 (2005), available at <http://www.craigball.com/metadata.pdf> (last visited Feb. 10, 2015).
- [5] I. Cho, H.-A. Park, Development and evaluation of a terminology-based electronic nursing record system, *Journal of Biomedical Informatics* 36 (2003) 304-312.
- [6] O. Bodenreider, B. Smith, A. Kumar, A. Burgun, Investigating Subsumption in DL-based Terminologies: A case Study in SNOMED-CT, *First International Workshop on Formal Biomedical Knowledge Representation (KR-MED 2004)*, (2004) 12-20.
- [7] G. Booch, *Object-oriented design with applications*, second ed., Redwood City: Benjamin/Cummings, 2004.
- [8] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, M. Stal, *Pattern-Oriented Software Architecture: A System Of Patterns*, West Sussex, England: John Wiley & Sons Ltd, 1996.
- [9] T. Bürkle, *Klassifikation, Konzeption und Anwendung medizinischer Data Dictionaries*, Humanmedizin, Justus-Leibig-Universität Gießen, Gießen, 2000.
- [10] W. Ceusters, B. Smith, A unified framework for biomedical terminologies and ontologies, *Stud Health Technol Inform* 160 (2010) 1050-1054.
- [11] J.J. Cimino, Desiderata for controlled medical vocabularies in the twenty-first century, *Methods Inf Med* 37 (1998) 394-403.
- [12] R. Cornet, Information-content-based measures for the structure of terminological systems and for data recorded using these systems, *Stud Health Technol Inform* 160(2) (2010) 1075-1079.
- [13] R. Cornet, N.F. de Keizer, A. Abu-Hanna, A framework for characterizing terminological systems, *Methods Inf Med* 45 (2006) 253-266.

- [14] A. Eliens, *Principles of Object-Oriented Software Development*, Addison Wesley, Cambridge, 1995.
- [15] R. Elmasri, S.B. Navathe, *Fundamentals of Database Systems*, Addison Wesley Longman, Amsterdam, Netherlands, 2003.
- [16] R.C. Feng, K.J. Tseng, H.F. Yan, H.Y. Huang, P. Chang, A preliminary study on the use of clinical care classification in nursing documentation data sets, *Comput Methods Programs Biomed* 112(3) (2013) 713-719.
- [17] W. Gaus, *Dokumentations- und Ordnungslehre*, Springer, Berlin, 2005.
- [18] A.M. van Ginneken, H. Stam, E.M. van Mulligen, M. de Wilde, R. van Mastrigt, J.H. van Bommel, ORCA: the versatile CPR, *Methods Inf Med* 38 (1999) 332-338.
- [19] A.M. van Ginneken, Considerations for the representation of meta-data for the support of structured data entry, *Methods Inf Med* 42 (2003) 226-235.
- [20] R. Goertzen, Ein integriertes System zur computerunterstützten Entwicklung medizinischer Dokumentationssysteme unter besonderer Berücksichtigung von Integritätsbedingungen (InCoMe), Medizinische Fakultät, Universität Duisburg-Essen, Duisburg, 2005.
- [21] R. Goertzen, J. Stausberg, A grammar of integrity constraints in medical documentation systems, *Comput Methods Programs Biomed* 86 (2007) 93-102.
- [22] J. Goll, *Methoden und Architekturen der Softwaretechnik*, Vieweg+Teubner Verlag; Auflage: 2011 (20. April 2011), ISBN: 978-3-8348-1578-1.
- [23] J. Goll, M. Dausmann, *Architektur- und Entwurfsmuster der Softwaretechnik: Mit lauffähigen Beispielen in Java [Gebundene Ausgabe]*, Springer Vieweg; Auflage: 2013 (28. August 2013), ISBN-13: 978-3834824318.
- [24] C. González, BG Blobel, DM López, Ontology-based framework for electronic health records interoperability, *Stud Health Technol Inform* 169 (2011) 694-698.
- [25] S. Herzberg, K. Rahbar, L. Stegger, M. Schafers, M. Dugas, Concept and implementation of a computer-based reminder system to increase completeness in clinical documentation, *Int J Med Inform* 80 (2011) 351-358.
- [26] K. Häyrynen, J. Lammintakanen, K. Saranto, Evaluation of electronic nursing documentation—Nursing process model and standardized terminologies as keys to visible and transparent nursing, *Int J Med Inform* 79 (2010) 554-564.
- [27] K. Häyrynen, K. Saranto, The use of nursing terminology in electronic documentation, *Stud Health Technol Inform* 146 (2009) 342-346.

- [28] J. Ingenerf, Terminologien oder Klassifikationen. Was bringt die Zukunft? , Bundesgesundheitsbl - Gesundheitsforsch - Gesundheitsschutz 50 (2007) 1070-1083.
- [29] J. Ingenerf, W. Giere, Concept-oriented standardization and statistics-oriented classification: continuing the classification versus nomenclature controversy, *Methods Inf Med* 37 (1998) 527-539.
- [30] E. Ingenhoven, J. Becker, Qualitätssicherung: Das Projekt Qualis - Medizinisches Dokumentationssystem, *Dtsch Arztebl Int* 97 (2000) 26-28.
- [31] ISO/IEC CD 11179-3.2 Information technology — Metadata registries (MDR) — Part 3: Registry Metamodel and basic attributes, 3rd Edition. ISO/IEC JTC 1/SC 32 N 1851, Date: 2009-03-23 Available at [http://jtc1sc32.org/doc/N1851-1900/32N1851Ta-text\\_for\\_ballot-CD2\\_11179-3.pdf](http://jtc1sc32.org/doc/N1851-1900/32N1851Ta-text_for_ballot-CD2_11179-3.pdf) [last access on 2012-12-12].
- [32] T. Itälä, J. Ukkola, A. Virtanen, J. Mykkänen, SOA approach for integration of departmental systems, *Stud Health Technol Inform* 136 (2008) 723-728.
- [33] S.B. Johnson, J.J. Cimino, C. Friedman, G. Hripcsak, P.D. Clayton, Using metadata to integrate medical knowledge in a clinical information system, *Proc Annu Symp Comput Appl Med Care* (1990) 340–344.
- [34] T. Jokinen, A. Ensio, K. Saranto, Using standardized terminology for nursing documentation, *Stud Health Technol Inform* 122 (2006) 995.
- [35] C. Kecher, *UML 2: Das umfassende Handbuch*, Vol. 3, Bonn: Galileo Computing, 2009.
- [36] T.Y. Kim, A. Coenen, N. Hardiker, A quality improvement model for healthcare terminologies, *J Biomed Inform* 43 (2010) 1036-1043.
- [37] U.M. Kinnunen, K. Saranto, M. Miettinen, Effects of terminology based documentation on nursing, *Stud Health Technol Inform* 146 (2009) 332-336.
- [38] P. Knaup-Gregori, S. Garde, R. Haux, Systematic planning of patient records for cooperative care and multicenter research, *Int J Med Inform* 76 (2007) 109-117.
- [39] G.A. Komatsoulis, D.B. Warzel, F.W. Hartel, K. Shanbhag, R. Chilukuri, G. Fragoso, S. Coronado, D.M. Reeves, J.B. Hadfield, C. Ludet, P.A. Covitz, caCORE version 3: Implementation of a model driven, service-oriented architecture for semantic interoperability, *J Biomed Inform* 41 (2008) 106-123.
- [40] R. Krumm, A. Semjonow, J. Tio, H. Duhme, T. Bürkle, J. Haier, M. Dugas, B. Breil, The need for harmonized structured documentation and chances of secondary use - results of a systematic analysis with automated form comparison for prostate and breast cancer, *J Biomed Inform* 51 (2014) 86-99.

- [41] T. Lehmann, Handbuch der Medizinischen Informatik, 2. Auflage, Carl Hanser Verlag München, Wien, 2005. ISBN 3-446-22701-6
- [42] F. Leiner, W. Gaus, R. Haux, P. Knaup-Gregori, Medizinische Dokumentation: Lehrbuch und Leitfaden für die Praxis, Schattauer, Stuttgart, 1999.
- [43] F. Leiner, W. Gaus, R. Haux, P. Knaup-Gregori, K.-P. Pfeiffer, Medizinische Dokumentation: Grundlagen einer qualitätsgesicherten integrierten Krankenversorgung. Lehrbuch und Leitfaden, 5. Auflage, Schattauer Verlag, Stuttgart, 2006.
- [44] F. Leiner, R. Haux, Systematic planning of clinical documentation, *Methods Inf Med* 35 (1996) 25-34.
- [45] F. Lum, Why is a terminology important?, *American Academy of Ophthalmology*, San Francisco, California 112 (2005) 173-174.
- [46] P. Marcus, SS. Braman, International classification of disease coding for obstructive lung disease: does it reflect appropriate clinical documentation?, *Chest* 138(1) (2010) 188-192.
- [47] A. Merzweiler, R. Weber, S. Garde, R. Haux, P. Knaup-Gregori, TERMTrial--terminology-based documentation systems for cooperative clinical trials, *Computer Methods and Programs in Biomedicine* 78 (2005) 11-24.
- [48] P.W. Moorman, A.M. Van Ginneken, J. Van der Lei, J.H. Van Bommel, A Model for structured Data Entry based on Explicit Descriptive knowledge, *Methods of Information in Medicine* 33 (1994) 454-463
- [49] S. Ngouongo, J. Stausberg, Abbildung von Klassifikationen auf den ISO/IEC Standard für Metadata Registries: Ein Beitrag zur Harmonisierung von Datenbeständen der medizinischen Forschung. In: P. Schmücker, K.-H. Ellsäcker, S. Hayna, 55. GMDS-Jahrestagung. Abstractband. Köln: GMDS, (2010) 243-244.
- [50] S. MN Ngouongo, J. Stausberg, Harmonisierung von Datenbeständen der Medizin - Ein Weg zu besserer Qualität und höherer Effizienz empirischer Forschung, *MDI* 2011, 13(3): 129-132.
- [51] S. MN Ngouongo, J. Stausberg, Integration of Classifications and Terminologies in Metadata Registries Based on ISO/IEC 11179, *Stud Health Technol Inform* 169 (2011) 744-748.
- [52] S. MN Ngouongo, J. Stausberg, Terminology-based documentation systems: a systemantic comparison of four different approaches, *Stud Health Technol Inform* 190 (2013) 97-99.

- [53] S. MN Nguoungo, M. Löbe, J. Stausberg, The ISO/IEC 11179 norm for metadata registries: does it cover healthcare standards in empirical research? *J Biomed Inform* 46(2) (2013) 318-327.
- [54] C. Ohmann, W. Kuchinke, Future developments of medical informatics from the viewpoint of networked clinical research. Interoperability and integration, *Methods Inf Med* 48 (2009) 45-54.
- [55] H.-A. Park, I. Cho, N. Byeun, Modeling a terminology-based electronic nursing record system: An object-oriented approach, *International Journal of Medical Informatics* 76 (2007) 735-746.
- [56] J. Pathak, J. Wang, S. Kashyap, M. Basford, R. Li, D.R. Masys, C.G. Chute, Mapping clinical phenotype data elements to standardized metadata repositories and controlled terminologies: the eMERGE Network experience, *J Am Med Inform Assoc* 18 (2011) 376-386.
- [57] R. Petrasch, O. Meimberg, *Model Driven Architecture. Eine Praxis orientierte Einführung in die MDA. Vol. 1*, Dpunkt Verlag, Heidelberg, 2006.
- [58] A. Rector, A. Rossi, M.F. Consorti, P. Zanstra, Practical development of re-usable terminologies: GALEN-IN-USE and the GALEN Organisation, *Int J Med Inform* 48 (1998) 71-84.
- [59] A.L. Rector, W.D. Solomon, W.A. Nowlan, T.W. Rush, P.E. Zanstra, W.M. Claassen, A Terminology Server for medical language and medical information systems, *Methods Inf Med* 34 (1995) 147-157.
- [60] R.L. Richesson, J.E. Andrews, J.P. Krischer, Use of SNOMED CT to Represent Clinical Research Data: A Semantic Characterization of Data Items on Case Report Forms in Vasculitis Research, *J Am Med Inform Assoc* 13 (2006) 536-546.
- [61] S.T. Rosenbloom, R.A. Miller, K.B. Johnson, P.L. Elkin, S.H. Brown, A Model for Evaluating Interface Terminologies, *Journal of the American Medical Informatics Association* 15 (2008) 65-76.
- [62] L. Sobin, M. Gospodarowicz, C. Wittekind, *TNM Classification of Malignant Tumours, Seventh Edition*, Chichester, Wiley-Blackwell, 2010.
- [63] J. Stausberg, U. Altmann, G. Antony, J. Drepper, U. Sax, A. Schütt, Registers for networked medical research in Germany. Situation and prospects, *Appl Clin Inf* 1 (2010) 408-418.
- [64] J. Stausberg, J. Hasford, Identification of Adverse Drug Events: The Use of ICD-10 Coded Diagnoses in Routine Hospital Data, *Dtsch Arztebl Int* 107 (2010) 23-29.

- [65] J. Stausberg, M. Lobe, P. Verplancke, J. Drepper, H. Herre, M. Loffler, Foundations of a metadata repository for databases of registers and trials, *Stud Health Technol Inform* 150 (2009) 409-413.
- [66] J. Stausberg, A. Wormek, U. Kraut, Terminological reference of a knowledge-based system: the data dictionary, *Medinfo* 8 (1995) 157-161.
- [67] J. Sternisko, *Biomedizinische Terminologiesysteme*, Albert-Ludwigs-Universität, Freiburg, 2009.
- [68] A. Thoroddsen, K. Saranto, A. Ehrenberg, W. Sermeus, Models, standards and structures of nursing documentation in European countries. *Stud Health Technol Inform* 146 (2009) 327-331.
- [69] S. Thun, *Medizinische Dokumentation und Kommunikation*, in: C. Johner, P. Haas (Hrsg.), *Praxishandbuch IT im Gesundheitswesen - Erfolgreich einführen, entwickeln, anwenden und betreiben*, Carl Hanser Verlag, München (2009) 131–160.
- [70] H. Tseng, S. Moorhead, The use of standardized terminology to represent nursing knowledge: nursing interventions relevant to safety for patients with cancer, *Stud Health Technol Inform* 201 (2014) 298-303.
- [71] G. Wade, S.T. Rosenbloom, The impact of SNOMED CT revisions on a mapped interface terminology: Terminology development and implementation issues, *Journal of Biomedical Informatics* 42 (2009) 490-493.

K ANHANG

K.1 MODELDIAGRAMME

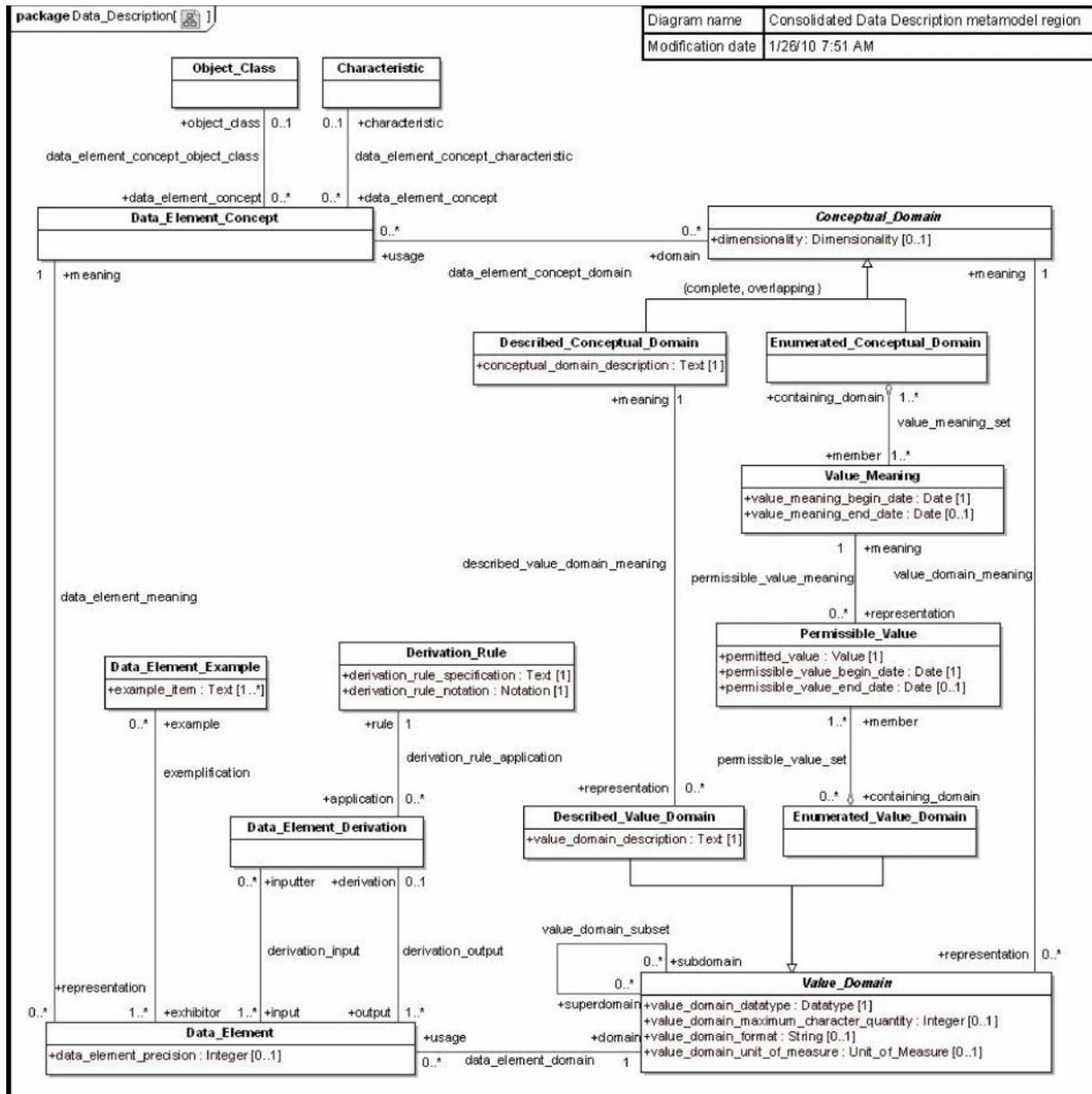


Abbildung 63: Consolidated Data Description Metamodel [31]

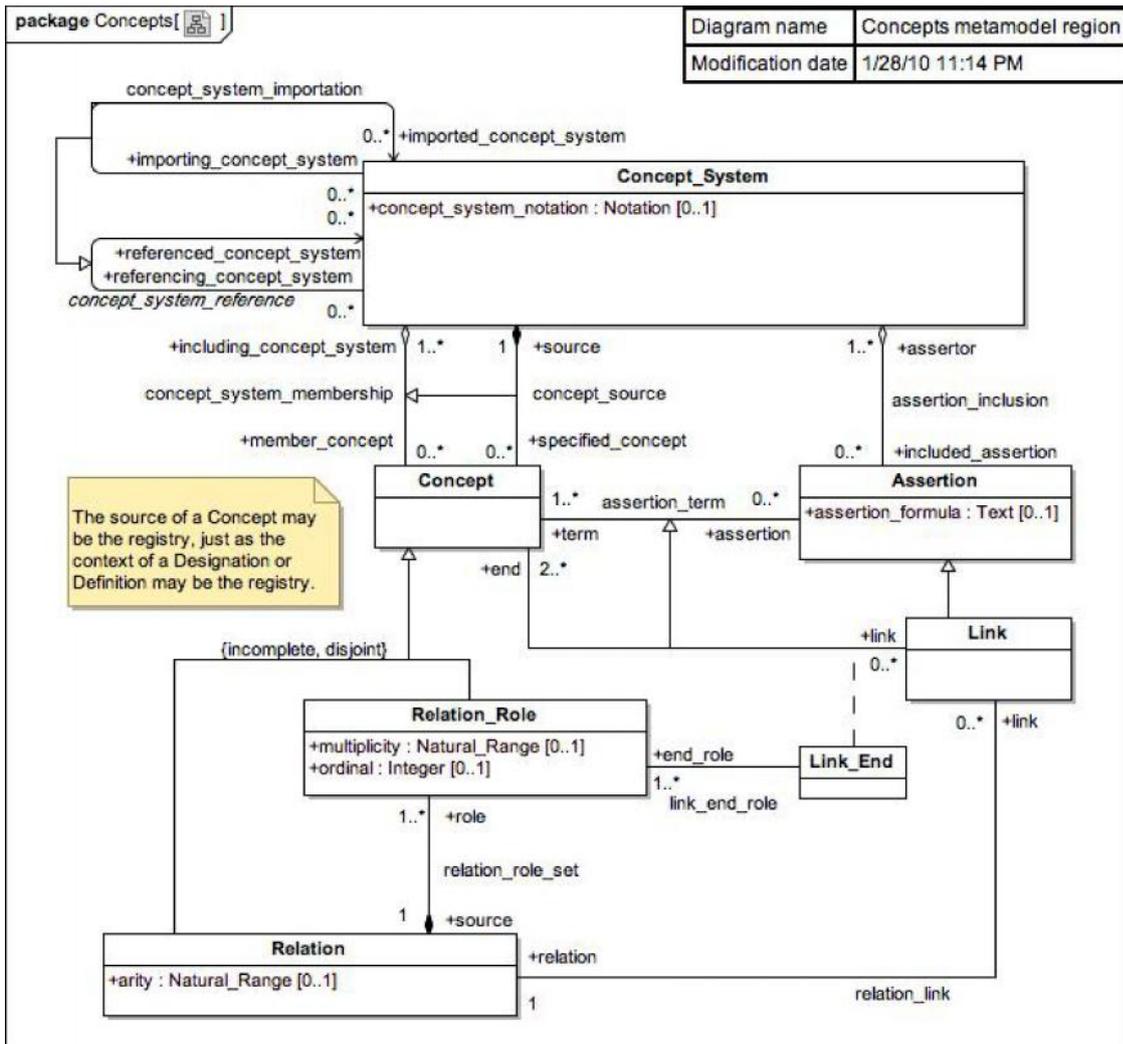


Abbildung 64: Concept system metamodel region [31]

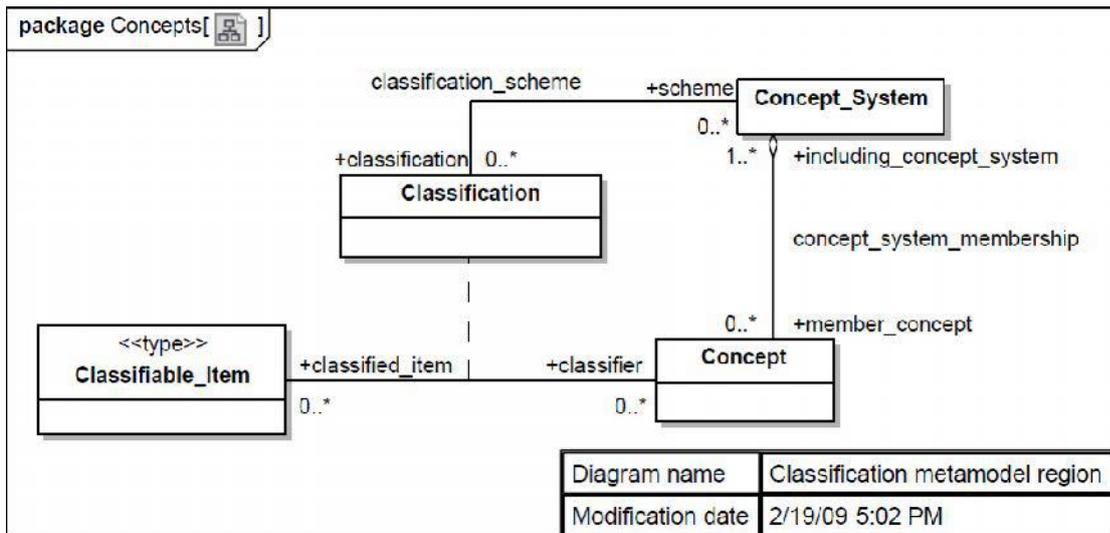


Abbildung 65: Classification metamodel region [31]

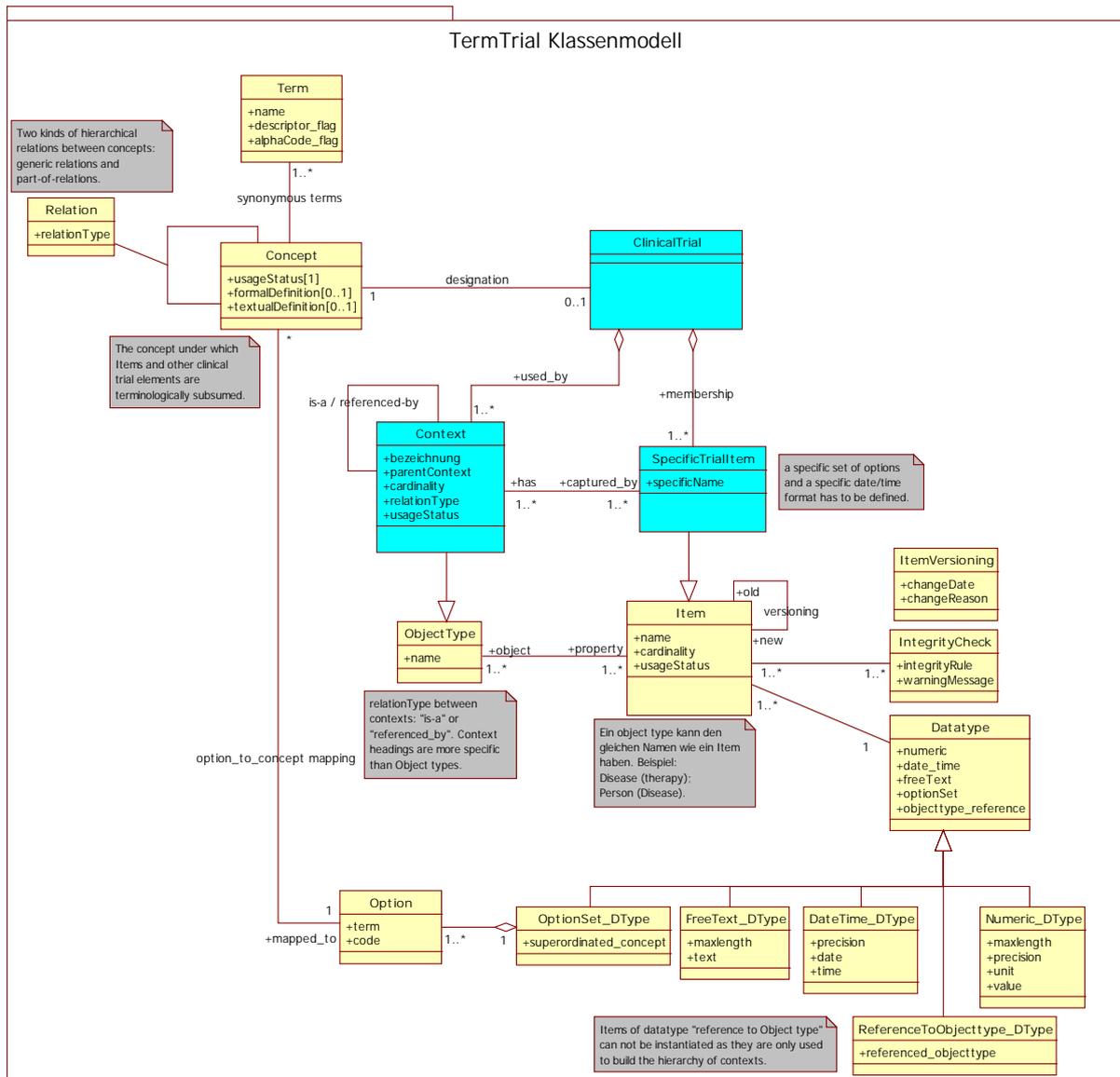
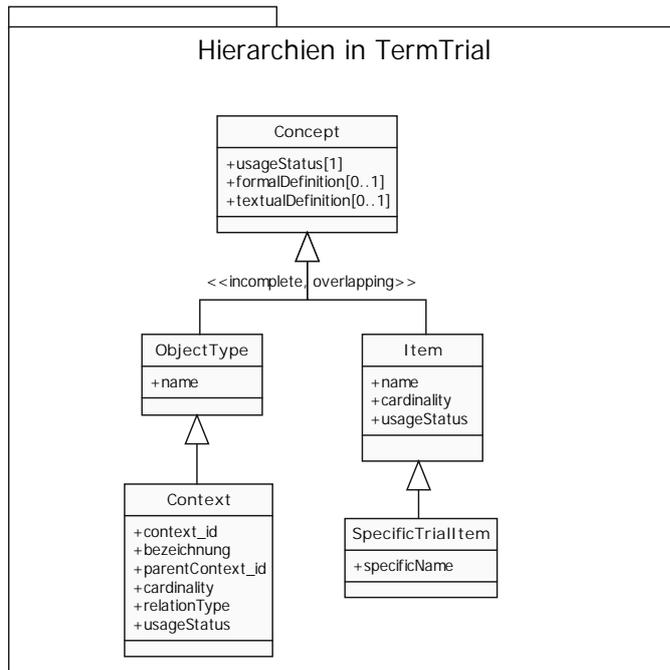
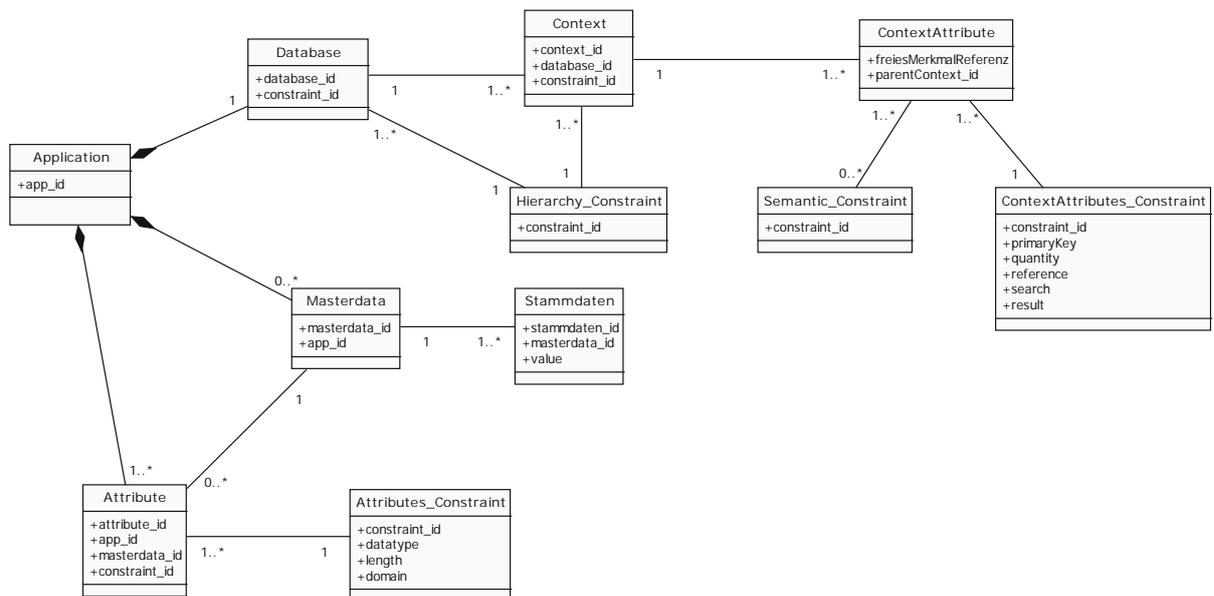


Abbildung 66: Vereinfachtes TERMTrial Klassenmodell [Ableitung aus 47]



**Abbildung 67: Klassenhierarchien in TERMTrial [Ableitung aus 47]**



**Abbildung 68: Vereinfachtes Klassendiagramm von InCoMe [Ableitung aus 20, 21]**

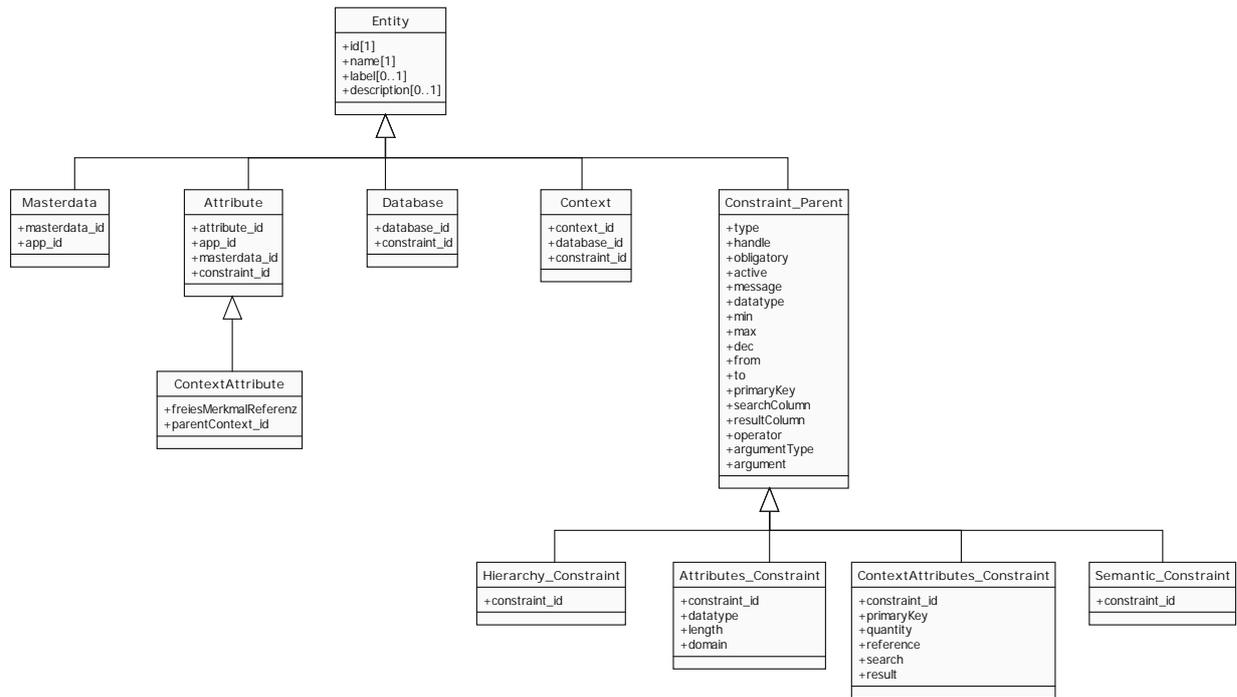


Abbildung 69: Modellelemente und deren Hierarchien in InCoMe [20, 21]

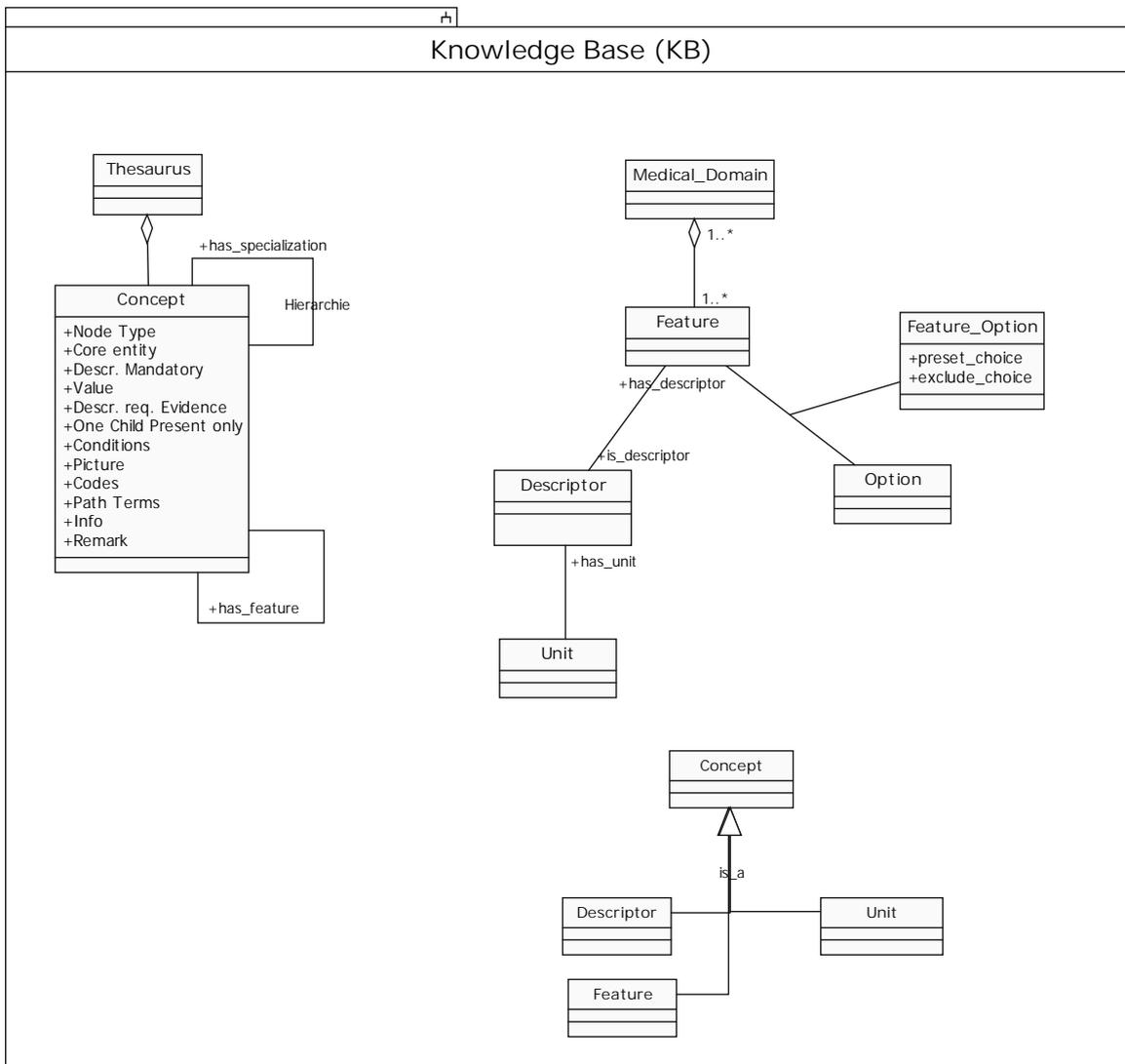


Abbildung 70: Vereinfachtes Datenmodell von ORCA [Ableitung aus 18, 48]

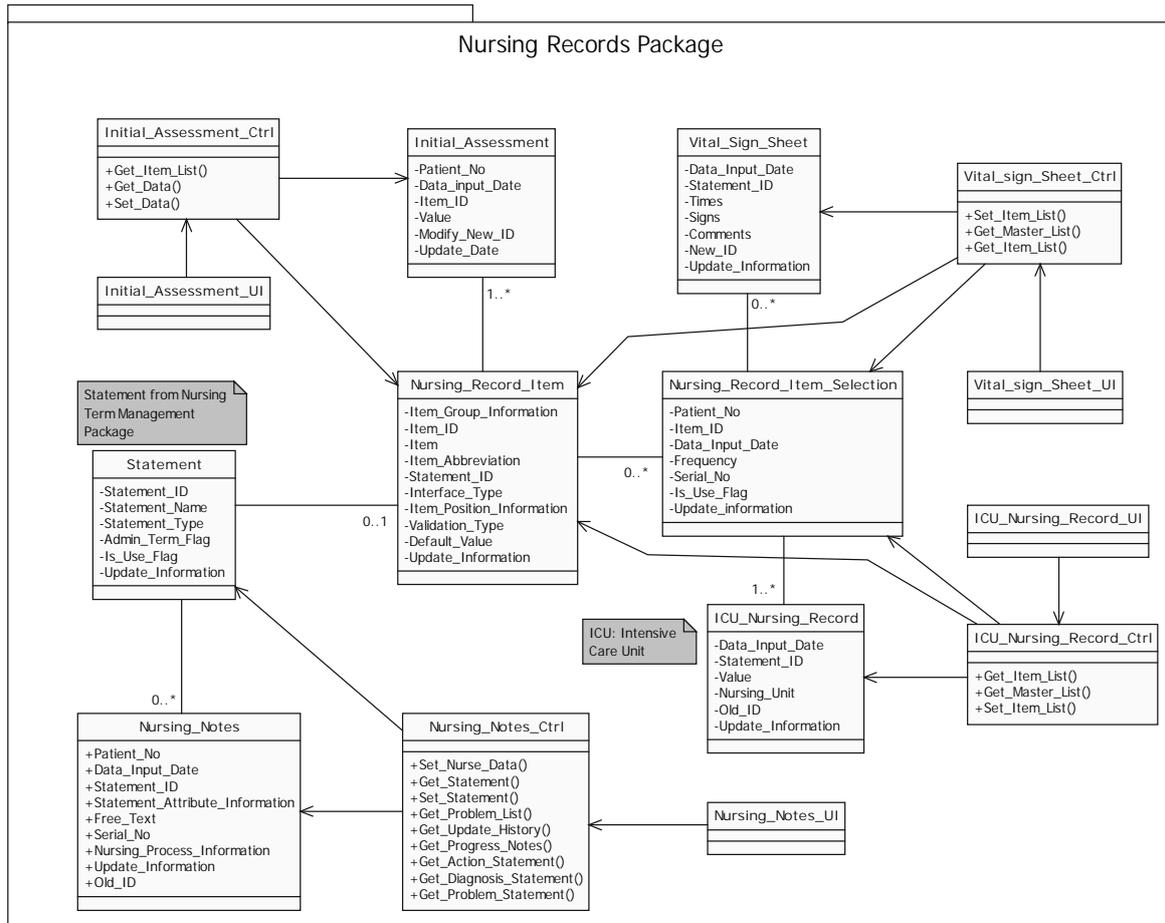


Abbildung 71: Klassendiagramm des Nursing Records Package [55]



- Anlage eines Schemas im verwendeten Datenbankmanagementsystem
  - Import des Datenbankskriptes in die Datenbank
  - Anlage einer Tabelle user\_admin zur Benutzerverwaltung mit den gewünschten Benutzern
4. Konfiguration der Anwendung (s. Abschnitt F.3.2.2)
    - Kopieren der Anwendungskonfigurationsdatei web.xml ins Anwendungsverzeichnis
    - Eintragung des Datenbanktreibers und der Datenbank-URL in die Anwendungskonfiguration
    - Speicherung der Applikationskonfiguration appl.xml ins Verzeichnis WEB-INF/classes/info der Webapplikation
  5. Starten des VokaDoks-Applikationsservers (s. Abschnitt F.3.2.3)

### ***K.2.2 SQL-Datenbankskript für die Erzeugung der Merkmalskatalog-Datenbank***

Nach Erzeugung der SQL-Datei aus dem VokaDoks-Designer und Installation des CDMS-Frameworks kann die dokumentationsspezifische Datenbank generiert werden. Im Folgenden wird das SQL-Skript beschrieben und angezeigt, das für diese Datenbankgenerierung zuständig ist (s. auch Abschnitt F.3.2.1).

Das SQL-Skript verfolgt folgende Struktur:

- Erzeugung eines Datenbankschemas, in dem sämtliche Daten gepflegt werden;
- Vergabe von Rechten auf die erzeugte CDMS-Datenbank an einem Administrator;
- Erzeugung sämtlicher Datenbank-Tabellen, in denen die Daten gespeichert werden, die später in das Dokumentationssystem eingetragen werden. Außer der Tabelle user\_admin, die für die Speicherung aller Benutzer des Dokumentationssystems zuständig ist, stellen alle anderen Tabellen die im VokaDoks-Designer erzeugten Kontexte mit ihren gebundenen Merkmalsarten als Tabellenspalten dar. Die modellierten Constraints von den Merkmalsarten werden hier als Eigenschaften bzw. Integritätsbedingungen von Tabellenspalten angezeigt.
- Anschließend werden die Stammdaten in ihren entsprechenden Tabellen eingefügt und Integritätsbedingungen zwischen Tabellen erzeugt.

```

SET FOREIGN_KEY_CHECKS = 0;
DROP DATABASE IF EXISTS 'merkmalskatalog';
CREATE DATABASE IF NOT EXISTS 'merkmalskatalog'
CHARACTER SET utf8 COLLATE utf8_general_ci;
GRANT ALL PRIVILEGES ON 'merkmalskatalog'.* TO ngouongo@localhost IDENTIFIED BY
'vokadoks';
FLUSH PRIVILEGES;
USE Merkmalskatalog;

DROP TABLE IF EXISTS user_admin;

```

```
CREATE TABLE user_admin (username VARCHAR(20) NOT NULL, pw VARCHAR(10) NOT NULL, PRIMARY KEY (username));
INSERT INTO user_admin VALUES ('admin', 'cdms');
INSERT INTO user_admin VALUES ('juergen', 'juergen');
GRANT ALL PRIVILEGES ON 'merkmalskatalog1'.* TO admin@localhost IDENTIFIED BY 'cdms';
GRANT ALL PRIVILEGES ON 'merkmalskatalog1'.* TO juergen@localhost IDENTIFIED BY 'juergen';
FLUSH PRIVILEGES;

DROP TABLE IF EXISTS Patient;
CREATE TABLE Patient (Patienten_ID INT NOT NULL, Erfassungsdatum DATE NOT NULL, Geburtsdatum DATE NOT NULL, Geschlecht INT (10), Anschrift VARCHAR (200), Groesse INT (3), Einheit VARCHAR (20), Schwangerschaft INT (10), PRIMARY KEY (Patienten_ID));
DROP TABLE IF EXISTS Visite;
CREATE TABLE Visite (ID INT NOT NULL, Patienten_ID INT NOT NULL, Erfassungsdatum DATE NOT NULL, Hauptdiagnose INT (10) NOT NULL, Anamnese VARCHAR (255), Nebendiagnose INT (10), Trauma INT (10), Allergien INT (10), Kostenträger INT (10) NOT NULL, PRIMARY KEY (ID));
DROP TABLE IF EXISTS Sozialdaten;
CREATE TABLE Sozialdaten (ID INT NOT NULL, Patienten_ID INT NOT NULL, Erfassungsdatum DATE NOT NULL, Muttersprache VARCHAR (100), Nationalität INT (10), PRIMARY KEY (ID));

DROP TABLE IF EXISTS Kostenträger;
CREATE TABLE Kostenträger (id INT (10) NOT NULL, value VARCHAR (255), PRIMARY KEY (id));
DROP TABLE IF EXISTS Ja_Nein_Auswahl;
CREATE TABLE Ja_Nein_Auswahl (id INT (10) NOT NULL, value VARCHAR (255), PRIMARY KEY (id));
DROP TABLE IF EXISTS JNU_Auswahl;
CREATE TABLE JNU_Auswahl (id INT (10) NOT NULL, value VARCHAR (255), PRIMARY KEY (id));
DROP TABLE IF EXISTS Geschlecht;
CREATE TABLE Geschlecht (id INT (10) NOT NULL, value VARCHAR (255), PRIMARY KEY
```

```
(id));
DROP TABLE IF EXISTS Hauptdiagnose;
CREATE TABLE Hauptdiagnose (id INT (10) NOT NULL, value VARCHAR (255), PRIMARY
KEY (id));
DROP TABLE IF EXISTS Nebendiagnose;
CREATE TABLE Nebendiagnose (id INT (10) NOT NULL, value VARCHAR (255), PRIMARY
KEY (id));
DROP TABLE IF EXISTS Nationalität;
CREATE TABLE Nationalität (id INT (10) NOT NULL, value VARCHAR (255), PRIMARY KEY
(id));

ALTER TABLE Patient ADD CONSTRAINT Patient_Geschlecht_FK FOREIGN KEY(Geschlecht)
REFERENCES Geschlecht(id);
ALTER TABLE Patient ADD CONSTRAINT Patient_Schwangerschaft_FK FOREIGN
KEY(Schwangerschaft) REFERENCES JNU_Auswahl(id);
ALTER TABLE Visite ADD CONSTRAINT Visite_Patient_FK FOREIGN KEY(Patienten_ID)
REFERENCES Patient(Patienten_ID);
ALTER TABLE Visite ADD CONSTRAINT Visite_Hauptdiagnose_FK FOREIGN
KEY(Hauptdiagnose) REFERENCES Hauptdiagnose(id);
ALTER TABLE Visite ADD CONSTRAINT Visite_Nebendiagnose_FK FOREIGN
KEY(Nebendiagnose) REFERENCES Nebendiagnose(id);
ALTER TABLE Visite ADD CONSTRAINT Visite_Trauma_FK FOREIGN KEY(Trauma)
REFERENCES Ja_Nein_Auswahl(id);
ALTER TABLE Visite ADD CONSTRAINT Visite_Allergien_FK FOREIGN KEY(Allergien)
REFERENCES Ja_Nein_Auswahl(id);
ALTER TABLE Visite ADD CONSTRAINT Visite_Kostenträger_FK FOREIGN
KEY(Kostenträger) REFERENCES Kostenträger(id);
ALTER TABLE Sozialdaten ADD CONSTRAINT Sozialdaten_Patient_FK FOREIGN
KEY(Patienten_ID) REFERENCES Patient(Patienten_ID);
ALTER TABLE Sozialdaten ADD CONSTRAINT Sozialdaten_Nationalität_FK FOREIGN
KEY(Nationalität) REFERENCES Nationalität(id);

INSERT INTO Kostenträger VALUES (1,'AOK');
INSERT INTO Kostenträger VALUES (2,'TK');
INSERT INTO Kostenträger VALUES (3,'Barmer GEK');
INSERT INTO Kostenträger VALUES (4,'BKK');
```

```
INSERT INTO Kostenträger VALUES (5,'IKK');
INSERT INTO Kostenträger VALUES (6,'Allianz');
INSERT INTO Ja_Nein_Auswahl VALUES (1,'Ja');
INSERT INTO Ja_Nein_Auswahl VALUES (2,'Nein');
INSERT INTO JNU_Auswahl VALUES (1,'Ja');
INSERT INTO JNU_Auswahl VALUES (2,'Nein');
INSERT INTO JNU_Auswahl VALUES (3,'Unbekannt');
INSERT INTO Geschlecht VALUES (1,'männlich');
INSERT INTO Geschlecht VALUES (2,'weiblich');
INSERT INTO Hauptdiagnose VALUES (1,'Keuchhusten');
INSERT INTO Hauptdiagnose VALUES (2,'Husten');
INSERT INTO Hauptdiagnose VALUES (3,'Notwendigkeit der Impfung gegen Keuchhusten
[Pertussis], nicht kombiniert');
INSERT INTO Hauptdiagnose VALUES (4,'Husten vermehrt');
INSERT INTO Hauptdiagnose VALUES (5,'Husten, Asthma gleichend');
INSERT INTO Hauptdiagnose VALUES (6,'trockener Husten');
INSERT INTO Hauptdiagnose VALUES (7,'Reizhusten');
INSERT INTO Hauptdiagnose VALUES (8,'Husten durch Allergien (nicht standardisierter Begriff)');
INSERT INTO Nebendiagnose VALUES (1,'Rückfallfieber <und> Fleckfieber');
INSERT INTO Nebendiagnose VALUES (2,'Q-Fieber <mit> Fieberkrämpfe');
INSERT INTO Nebendiagnose VALUES (3,'Gelbfieber <ohne> Hämorrhagisches Fieber durch
Arenaviren');
INSERT INTO Nationalität VALUES (1,'Kamerun');
INSERT INTO Nationalität VALUES (2,'Kenya');
INSERT INTO Nationalität VALUES (3,'Kongo');
INSERT INTO Nationalität VALUES (4,'Deutschland');
INSERT INTO Nationalität VALUES (5,'Frankreich');
INSERT INTO Nationalität VALUES (6,'England');
INSERT INTO Nationalität VALUES (7,'Italien');
INSERT INTO Nationalität VALUES (8,'Schweiz');
INSERT INTO Nationalität VALUES (9,'Eden');

SET FOREIGN_KEY_CHECKS = 1;
```

### K.2.3 Anwendung appl.xml

Im Folgenden wird die Konfigurationsdatei appl.xml für das VokaDoks-Dokumentationssystem als Webanwendung im Webserver beschrieben (s. Abschnitt F.3.2.2). Die appl.xml Konfigurationsdatei ist eine XML-Beschreibung der Struktur der Dokumentationssystem-Anwendung und beschreibt in welcher Reihenfolge die Datenfelder anzuzeigen sind bzw. unter welcher Hierarchischen Struktur sie miteinander stehen (Darstellungslogik). Zu jedem Datenfeld ist eine bestimmte Anzahl von Integritätsbedingungen hinterlegt.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<application name="Merkmalskatalog" description="IBE Merkmalskatalog">
<constraint type="hierarchy">
  <child context="Patient" />
</constraint>
<context name="Patient" label="Patient">
  <constraint type="hierarchy">
    <child context="Visite" />
    <child context="Sozialdaten" />
  </constraint>
<attribute name="Patienten_ID" label="Patienten_ID" unityFlag="no" unityValue="" >
  <constraint type="key" primaryKey="true" />
  <constraint type="quantity" min="1" max="1" message="Falsche Anzahl" />
  <constraint type="reference" />
  <constraint type="search" searchColumn="true" />
  <constraint type="result" resultColumn="true" />
  <constraint type="type" datatype="integer" message="Falscher Datentyp" />
  <constraint type="length" min="1" max="6" dec="0" message="Falsche Feldlänge" />
  <constraint type="domain" from="0" to="888888" message="Wert außerhalb des Wertebereichs" />
  <constraint type="unity" message="Messeinheit ungültig"/>
</attribute>
<attribute name="Erfassungsdatum" label="Erfassungsdatum" unityFlag="no" unityValue="" >
  <constraint type="key" primaryKey="false" />
  <constraint type="quantity" min="1" max="1" message="Falsche Anzahl" />
  <constraint type="reference" />
  <constraint type="search" searchColumn="true" />
  <constraint type="result" resultColumn="true" />
  <constraint type="semantic" active="true" handle="error" message="Erfassungsdatum muss
```

```

größer als Geburtsdatum sein!" operator="GT" argumenttype="reference"
argument="Patient.Geburtsdatum" />
  <constraint type="type" datatype="date" message="Falscher Datentyp" />
  <constraint type="length" min="0" max="0" dec="0" message="Falsche Feldlänge" />
  <constraint type="domain" message="Wert außerhalb des Wertebereichs" />
  <constraint type="unity" message="Messeinheit ungültig"/>
</attribute>
<attribute name="Geburtsdatum" label="Geburtsdatum" unityFlag="no" unityValue="" >
  <constraint type="key" primaryKey="false" />
  <constraint type="quantity" min="1" max="1" message="Falsche Anzahl" />
  <constraint type="reference" />
  <constraint type="search" searchColumn="true" />
  <constraint type="result" resultColumn="true" />
  <constraint type="semantic" active="true" handle="error" message="Erfassungsdatum muss
größer als Geburtsdatum sein!" operator="LT" argumenttype="reference"
argument="Patient.Erfassungsdatum" />
  <constraint type="type" datatype="date" message="Falscher Datentyp" />
  <constraint type="length" min="0" max="0" dec="0" message="Falsche Feldlänge" />
  <constraint type="domain" message="Wert außerhalb des Wertebereichs" />
  <constraint type="unity" message="Messeinheit ungültig"/>
</attribute>
<attribute name="Geschlecht" label="Geschlecht" unityFlag="no" unityValue="" >
  <constraint type="key" primaryKey="false" />
  <constraint type="quantity" min="0" max="1" message="Falsche Anzahl" />
  <constraint type="reference" />
  <constraint type="search" searchColumn="false" />
  <constraint type="result" resultColumn="true" />
  <constraint type="semantic" active="true" handle="error" message="Geschlecht muss bei
Schwangerschaft weiblich sein!" operator="EQ" argumenttype="value" argument="2">
  <and>
    <statement argument1="Patient.Schwangerschaft" operator="EQ" argumenttype="value"
argument2="1" />
  </and>
</constraint>
<constraint type="type" datatype="reference" message="Falscher Datentyp" />
<constraint type="length" min="0" max="0" dec="0" message="Falsche Feldlänge" />

```

```

<constraint type="domain" message="Wert außerhalb des Wertebereichs">
  <foreign-key foreignTable="Geschlecht">
    <reference local="Geschlecht" foreign="id" />
  </foreign-key>
</constraint>
<constraint type="unity" message="Messeinheit ungültig"/>
</attribute>
<attribute name="Anschrift" label="Anschrift" unityFlag="no" unityValue="" >
  <constraint type="key" primaryKey="false" />
  <constraint type="quantity" min="0" max="1" message="Falsche Anzahl" />
  <constraint type="reference" />
  <constraint type="search" searchColumn="false" />
  <constraint type="result" resultColumn="false" />
  <constraint type="type" datatype="character" message="Falscher Datentyp" />
  <constraint type="length" min="0" max="200" dec="0" message="Falsche Feldlänge" />
  <constraint type="domain" message="Wert außerhalb des Wertebereichs" />
  <constraint type="unity" message="Messeinheit ungültig"/>
</attribute>
<attribute name="Größe" label="Größe" unityFlag="yes" unityValue="cm" >
  <constraint type="key" primaryKey="false" />
  <constraint type="quantity" min="0" max="1" message="Falsche Anzahl" />
  <constraint type="reference" />
  <constraint type="search" searchColumn="false" />
  <constraint type="result" resultColumn="true" />
  <constraint type="type" datatype="number" message="Falscher Datentyp" />
  <constraint type="length" min="0" max="3" dec="0" message="Falsche Feldlänge" />
  <constraint type="domain" from="0" to="300" message="Wert außerhalb des Wertebereichs" />
  <constraint type="unity" message="Messeinheit ungültig" />
</attribute>
<attribute name="Einheit" label="Größeneinheit">
  <constraint type="key" primaryKey="false" />
  <constraint type="quantity" min="0" max="1" message="Falsche Anzahl" />
  <constraint type="reference" />
  <constraint type="search" searchColumn="false" />
  <constraint type="result" resultColumn="true" />
  <constraint type="type" datatype="character" message="Falscher Datentyp" />

```

```

<constraint type="length" min="0" max="20" dec="0" message="Falsche Feldlänge" />
<constraint type="domain" message="Wert außerhalb des Wertebereichs" />
<constraint type="unity" unity="cm" message="Messeinheit ungültig" />
</attribute>
<attribute name="Schwangerschaft" label="Schwangerschaft" unityFlag="no" unityValue="" >
  <constraint type="key" primaryKey="false" />
  <constraint type="quantity" min="0" max="1" message="Falsche Anzahl" />
  <constraint type="reference" />
  <constraint type="search" searchColumn="false" />
  <constraint type="result" resultColumn="true" />
  <constraint type="semantic" active="true" handle="error" message="Geschlecht muss bei
Schwangerschaft weiblich sein!" operator="EQ" argumenttype="value" argument="1">
    <and>
      <statement argument1="Patient.Geschlecht" operator="EQ" argumenttype="value"
argument2="2" />
    </and>
  </constraint>
  <constraint type="type" datatype="reference" message="Falscher Datentyp" />
  <constraint type="length" min="0" max="0" dec="0" message="Falsche Feldlänge" />
  <constraint type="domain" message="Wert außerhalb des Wertebereichs">
    <foreign-key foreignTable="JNU_Auswahl">
      <reference local="Schwangerschaft" foreign="id" />
    </foreign-key>
  </constraint>
  <constraint type="unity" message="Messeinheit ungültig"/>
</attribute>
</context>
<context name="Visite" label="Visite">
  <constraint type="hierarchy" />
  <attribute name="ID" label="ID" unityFlag="no" unityValue="" >
    <constraint type="key" primaryKey="true" />
    <constraint type="quantity" min="1" max="1" message="Falsche Anzahl" />
    <constraint type="reference"/>
    <constraint type="search" searchColumn="true" />
    <constraint type="result" resultColumn="true" />
    <constraint type="type" datatype="integer" message="Falscher Datentyp" />

```

```

    <constraint type="length" min="1" max="6" dec="0" message="Falsche Feldlänge" />
    <constraint type="domain" from="0" to="888888" message="Wert außerhalb des Wertebereichs"
/>
    <constraint type="unity" message="Messeinheit ungültig" />
</attribute>
<attribute name="Patienten_ID" label="Patienten_ID" unityFlag="no" unityValue="" >
    <constraint type="key" primaryKey="true" />
    <constraint type="quantity" min="1" max="1" message="Falsche Anzahl" />
    <constraint type="reference">
        <foreign-key foreignTable="Patient">
            <reference local="Patienten_ID" foreign="Patienten_ID" />
        </foreign-key>
    </constraint>
    <constraint type="search" searchColumn="true" />
    <constraint type="result" resultColumn="true" />
    <constraint type="type" datatype="integer" message="Falscher Datentyp" />
    <constraint type="length" min="1" max="6" dec="0" message="Falsche Feldlänge" />
    <constraint type="domain" from="0" to="888888" message="Wert außerhalb des Wertebereichs"
/>
    <constraint type="unity" message="Messeinheit ungültig" />
</attribute>
<attribute name="Erfassungsdatum" label="Erfassungsdatum" unityFlag="no" unityValue="" >
    <constraint type="key" primaryKey="false" />
    <constraint type="quantity" min="1" max="1" message="Falsche Anzahl" />
    <constraint type="reference" />
    <constraint type="search" searchColumn="true" />
    <constraint type="result" resultColumn="true" />
    <constraint type="type" datatype="date" message="Falscher Datentyp" />
    <constraint type="length" min="0" max="0" dec="0" message="Falsche Feldlänge" />
    <constraint type="domain" message="Wert außerhalb des Wertebereichs" />
    <constraint type="unity" message="Messeinheit ungültig" />
</attribute>
<attribute name="Hauptdiagnose" label="Hauptdiagnose" unityFlag="no" unityValue="" >
    <constraint type="key" primaryKey="false" />
    <constraint type="quantity" min="1" max="1" message="Falsche Anzahl" />
    <constraint type="reference" />

```

```
<constraint type="search" searchColumn="true" />
<constraint type="result" resultColumn="true" />
<constraint type="type" datatype="reference" message="Falscher Datentyp" />
<constraint type="length" min="0" max="0" dec="0" message="Falsche Feldlänge" />
<constraint type="domain" message="Wert außerhalb des Wertebereichs">
  <foreign-key foreignTable="Hauptdiagnose">
    <reference local="Hauptdiagnose" foreign="id" />
  </foreign-key>
</constraint>
<constraint type="unity" message="Messeinheit ungültig" />
</attribute>
<attribute name="Anamnese" label="Anamnese" unityFlag="no" unityValue="" >
  <constraint type="key" primaryKey="false" />
  <constraint type="quantity" min="0" max="1" message="Falsche Anzahl" />
  <constraint type="reference" />
  <constraint type="search" searchColumn="false" />
  <constraint type="result" resultColumn="false" />
  <constraint type="type" datatype="character" message="Falscher Datentyp" />
  <constraint type="length" min="0" max="255" dec="0" message="Falsche Feldlänge" />
  <constraint type="domain" message="Wert außerhalb des Wertebereichs" />
  <constraint type="unity" message="Messeinheit ungültig" />
</attribute>
<attribute name="Nebendiagnose" label="Nebendiagnose" unityFlag="no" unityValue="" >
  <constraint type="key" primaryKey="false" />
  <constraint type="quantity" min="0" max="1" message="Falsche Anzahl" />
  <constraint type="reference" />
  <constraint type="search" searchColumn="true" />
  <constraint type="result" resultColumn="true" />
  <constraint type="type" datatype="reference" message="Falscher Datentyp" />
  <constraint type="length" min="0" max="0" dec="0" message="Falsche Feldlänge" />
  <constraint type="domain" message="Wert außerhalb des Wertebereichs">
    <foreign-key foreignTable="Nebendiagnose">
      <reference local="Nebendiagnose" foreign="id" />
    </foreign-key>
  </constraint>
  <constraint type="unity" message="Messeinheit ungültig" />
```

```

</attribute>
<attribute name="Trauma" label="Hat der Patient eine Kopferschütterung?" unityFlag="no"
unityValue="" >
  <constraint type="key" primaryKey="false" />
  <constraint type="quantity" min="0" max="1" message="Falsche Anzahl" />
  <constraint type="reference" />
  <constraint type="search" searchColumn="false" />
  <constraint type="result" resultColumn="false" />
  <constraint type="type" datatype="reference" message="Falscher Datentyp" />
  <constraint type="length" min="0" max="0" dec="0" message="Falsche Feldlänge" />
  <constraint type="domain" message="Wert außerhalb des Wertebereichs">
    <foreign-key foreignTable="Ja_Nein_Auswahl">
      <reference local="Trauma" foreign="id" />
    </foreign-key>
  </constraint>
  <constraint type="unity" message="Messeinheit ungültig" />
</attribute>
<attribute name="Allergien" label="Hat der Patient Allergien?" unityFlag="no" unityValue="" >
  <constraint type="key" primaryKey="false" />
  <constraint type="quantity" min="0" max="1" message="Falsche Anzahl" />
  <constraint type="reference" />
  <constraint type="search" searchColumn="false" />
  <constraint type="result" resultColumn="false" />
  <constraint type="type" datatype="reference" message="Falscher Datentyp" />
  <constraint type="length" min="0" max="0" dec="0" message="Falsche Feldlänge" />
  <constraint type="domain" message="Wert außerhalb des Wertebereichs">
    <foreign-key foreignTable="Ja_Nein_Auswahl">
      <reference local="Allergien" foreign="id" />
    </foreign-key>
  </constraint>
  <constraint type="unity" message="Messeinheit ungültig" />
</attribute>
<attribute name="Kostentraeger" label="Kostenträger" unityFlag="no" unityValue="" >
  <constraint type="key" primaryKey="false" />
  <constraint type="quantity" min="1" max="1" message="Falsche Anzahl" />
  <constraint type="reference" />

```

```

<constraint type="search" searchColumn="true" />
<constraint type="result" resultColumn="true" />
<constraint type="type" datatype="reference" message="Falscher Datentyp" />
<constraint type="length" min="0" max="0" dec="0" message="Falsche Feldlänge" />
<constraint type="domain" message="Wert außerhalb des Wertebereichs">
  <foreign-key foreignTable="Kostentraeger">
    <reference local="Kostentraeger" foreign="id" />
  </foreign-key>
</constraint>
<constraint type="unity" message="Messeinheit ungültig" />
</attribute>
</context>
<context name="Sozialdaten" label="Sozialdaten">
  <constraint type="hierarchy" />
  <attribute name="ID" label="ID" unityFlag="no" unityValue="" >
    <constraint type="key" primaryKey="true" />
    <constraint type="quantity" min="1" max="1" message="Falsche Anzahl" />
    <constraint type="reference"/>
    <constraint type="search" searchColumn="true" />
    <constraint type="result" resultColumn="true" />
    <constraint type="type" datatype="integer" message="Falscher Datentyp" />
    <constraint type="length" min="1" max="6" dec="0" message="Falsche Feldlänge" />
    <constraint type="domain" from="0" to="888888" message="Wert außerhalb des Wertebereichs"
  />
  <constraint type="unity" message="Messeinheit ungültig" />
</attribute>
<attribute name="Patienten_ID" label="Patienten_ID" unityFlag="no" unityValue="" >
  <constraint type="key" primaryKey="true" />
  <constraint type="quantity" min="1" max="1" message="Falsche Anzahl" />
  <constraint type="reference">
    <foreign-key foreignTable="Patient">
      <reference local="Patienten_ID" foreign="Patienten_ID" />
    </foreign-key>
  </constraint>
  <constraint type="search" searchColumn="true" />
  <constraint type="result" resultColumn="true" />

```

```

<constraint type="type" datatype="integer" message="Falscher Datentyp" />
<constraint type="length" min="1" max="6" dec="0" message="Falsche Feldlänge" />
<constraint type="domain" from="0" to="888888" message="Wert außerhalb des Wertebereichs"
/>
<constraint type="unity" message="Messeinheit ungültig" />
</attribute>
<attribute name="Erfassungsdatum" label="Erfassungsdatum" unityFlag="no" unityValue="" >
<constraint type="key" primaryKey="false" />
<constraint type="quantity" min="1" max="1" message="Falsche Anzahl" />
<constraint type="reference" />
<constraint type="search" searchColumn="true" />
<constraint type="result" resultColumn="true" />
<constraint type="type" datatype="date" message="Falscher Datentyp" />
<constraint type="length" min="0" max="0" dec="0" message="Falsche Feldlänge" />
<constraint type="domain" message="Wert außerhalb des Wertebereichs" />
<constraint type="unity" message="Messeinheit ungültig" />
</attribute>
<attribute name="Muttersprache" label="Muttersprache" unityFlag="no" unityValue="" >
<constraint type="key" primaryKey="false" />
<constraint type="quantity" min="0" max="1" message="Falsche Anzahl" />
<constraint type="reference" />
<constraint type="search" searchColumn="false" />
<constraint type="result" resultColumn="false" />
<constraint type="type" datatype="character" message="Falscher Datentyp" />
<constraint type="length" min="0" max="100" dec="0" message="Falsche Feldlänge" />
<constraint type="domain" message="Wert außerhalb des Wertebereichs" />
<constraint type="unity" message="Messeinheit ungültig" />
</attribute>
<attribute name="Nationalitaet" label="Nationalität" unityFlag="no" unityValue="" >
<constraint type="key" primaryKey="false" />
<constraint type="quantity" min="0" max="1" message="Falsche Anzahl" />
<constraint type="reference" />
<constraint type="search" searchColumn="false" />
<constraint type="result" resultColumn="false" />
<constraint type="type" datatype="reference" message="Falscher Datentyp" />
<constraint type="length" min="0" max="0" dec="0" message="Falsche Feldlänge" />

```

```

<constraint type="domain" message="Wert außerhalb des Wertebereichs">
  <foreign-key foreignTable="Nationalitaet">
    <reference local="Nationalitaet" foreign="id" />
  </foreign-key>
</constraint>
<constraint type="unity" message="Messeinheit ungültig" />
</attribute>
</context>
</application>

```

#### ***K.2.4 Anwendungskonfiguration web.xml***

Im Folgenden wird die Konfigurationsdatei web.xml für das VokaDoks-Dokumentationssystem als Webanwendung im Webserver beschrieben (s. Abschnitt F.3.2.2). Java-Webanwendungen verwenden eine Konfigurationsdatei namens web.xml, um Benutzer-Authentifizierungen zu prüfen, Verbindungsparameter zur Anwendungsdatenbank einzustellen und vor allem zu beschreiben wie eine Webanwendung in einem Webserver wie Tomcat anzuzeigen bzw. bereitzustellen ist, sogenannte Business-Logik.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app
PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
"http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">

<web-app>
<display-name>VokaDoks Merkmalskatalog</display-name>
  <description>
    Welcome to VokaDoks Dokumentationssystem
  </description>
  <servlet>
    <servlet-name>authenticate</servlet-name>
    <servlet-class>servlets.LoginServlet</servlet-class>
    <init-param>
      <param-name>check</param-name>
      <param-value>>true</param-value>
    </init-param>
    <init-param>
      <param-name>dbDriver</param-name>

```

```
<param-value>com.mysql.jdbc.Driver</param-value>
</init-param>
<init-param>
  <param-name>dbUrl</param-name>
  <param-value>jdbc:mysql://localhost:3306/merkmalskatalog</param-value>
</init-param>
<init-param>
  <param-name>dbUser</param-name>
  <param-value>juergen</param-value>
</init-param>
<init-param>
  <param-name>dbPass</param-name>
  <param-value>juergen</param-value>
</init-param>
</servlet>
<servlet>
  <servlet-name>patient</servlet-name>
  <servlet-class>servlets.CDMSServlet</servlet-class>
  <init-param>
    <param-name>action</param-name>
    <param-value>search</param-value>
  </init-param>
  <init-param>
    <param-name>context</param-name>
    <param-value>Patient</param-value>
  </init-param>
</servlet>
<servlet>
  <servlet-name>visite</servlet-name>
  <servlet-class>servlets.CDMSServlet</servlet-class>
  <init-param>
    <param-name>action</param-name>
    <param-value>search</param-value>
  </init-param>
  <init-param>
    <param-name>context</param-name>
```

```
        <param-value>Visite</param-value>
    </init-param>
</servlet>
<servlet>
    <servlet-name>sozialdaten</servlet-name>
    <servlet-class>servlets.CDMSServlet</servlet-class>
    <init-param>
        <param-name>action</param-name>
        <param-value>search</param-value>
    </init-param>
    <init-param>
        <param-name>context</param-name>
        <param-value>Sozialdaten</param-value>
    </init-param>
</servlet>
<servlet>
    <servlet-name>LogoutServlet</servlet-name>
    <servlet-class>servlets.LogoutServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>authenticate</servlet-name>
    <url-pattern>/authenticate</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>patient</servlet-name>
    <url-pattern>/Patient</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>visite</servlet-name>
    <url-pattern>/Visite</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>sozialdaten</servlet-name>
    <url-pattern>/Sozialdaten</url-pattern>
</servlet-mapping>
<servlet-mapping>
```

```

        <servlet-name>LogoutServlet</servlet-name>
        <url-pattern>/LogoutServlet</url-pattern>
</servlet-mapping>
<!-- The Welcome File List -->
<welcome-file-list>
    <welcome-file>introduction.jsp</welcome-file>
</welcome-file-list>
<!-- Struts Tag Library Descriptors -->
<taglib>
    <taglib-uri>/WEB-INF/struts-bean.tld</taglib-uri>
    <taglib-location>/WEB-INF/struts-bean.tld</taglib-location>
</taglib>
<taglib>
    <taglib-uri>/WEB-INF/struts-html.tld</taglib-uri>
    <taglib-location>/WEB-INF/struts-html.tld</taglib-location>
</taglib>
<taglib>
    <taglib-uri>/WEB-INF/struts-logic.tld</taglib-uri>
    <taglib-location>/WEB-INF/struts-logic.tld</taglib-location>
</taglib>
<taglib>
    <taglib-uri>/WEB-INF/struts-template.tld</taglib-uri>
    <taglib-location>/WEB-INF/struts-template.tld</taglib-location>
</taglib>
</web-app>

```

### ***K.2.5 CDMS-Source code und Anwendung***

Sämtliche Klassen bzw. Kodierungsdateien des VokaDoks-Designers und der VokaDoks-basierten CDMS-Anwendung werden als Anlage zu der vorliegenden Arbeit zur Verfügung gestellt.



## DANKSAGUNG

---

An erster Stelle möchte ich mich ganz herzlich bei meinem wissenschaftlichen Betreuer Prof. Dr. Jürgen Stausberg für die Themenstellung dieser Arbeit, für seine zahlreichen Anregungen und unsere stets hilfreichen Diskussionen, und ganz besonders für seine Geduld bedanken. Dafür möchte ich mich zudem bei ihm ganz speziell bedanken, dass er mich als wissenschaftliche Mitarbeiterin für das ISO-11179-Projekt im Institut für Medizinische Informatik, Biometrie und Epidemiologie (IBE) der Ludwig-Maximilians-Universität München eingestellt hat, obwohl die Konkurrenz eigentlich unschlagbar schien. Ich werde dem Prof. Stausberg für immer und ewig dankbar sein.

Nicht zuletzt danke ich meinen Eltern, Anne Nguenpang und Emmanuel Nguemegne, die mir das Licht des Lebens erblicken lassen haben, und ganz speziell meinem Vater für seine langjährige Unterstützung und seine Anregung mich weiter wissenschaftlich fortzubilden.

Meinen Geschwistern, Gertrude und Ghislain, möchte ich ebenfalls ganz besonders für ihre wiederholte und anregende bis nervige Frage „Bist du schon mit deiner Doktorarbeit fertig?“ danken.

Ich möchte mich schließlich bei allen bedanken, deren Begegnung oder Unterstützung oder Worte das Entstehen bzw. die Fertigstellung dieser Arbeit stark begünstigt hat.

*"Für alles bin ich stark durch den [**Jehova**], der mir Kraft verleiht." Philipper 4:13*

*„Möge der Name Gottes gesegnet werden von unabsehbarer Zeit ja bis auf unabsehbare Zeit, denn Weisheit und Macht — denn sie gehören ihm. [...]*

*Dich, o Gott meiner Vorväter, preise und lobe ich, weil du mir Weisheit und Macht gegeben hast. [...]"*

*Daniel 2:19-23*

*„Du bist würdig, **Jehova**, ja du, unser Gott, die Herrlichkeit und die Ehre und die Macht zu empfangen, weil du alle Dinge erschaffen hast, und deines Willens wegen existierten sie und wurden sie erschaffen.“*

*Offenbarung 4:11*

## Eidesstattliche Versicherung

**Nguoungo Nguemegne, Sylvie Marie-Noel**

---

Name, Vorname

Ich erkläre hiermit an Eides statt,

dass ich die vorliegende Dissertation mit dem Thema

**Terminologische Fundierung von Dokumentationssystemen - Ein Beitrag zur vernetzten Forschung und integrierten Versorgung -**

selbständig verfasst, mich außer der angegebenen keiner weiteren Hilfsmittel bedient und alle Erkenntnisse, die aus dem Schrifttum ganz oder annähernd übernommen sind, als solche kenntlich gemacht und nach ihrer Herkunft unter Bezeichnung der Fundstelle einzeln nachgewiesen habe.

Ich erkläre des Weiteren, dass die hier vorgelegte Dissertation nicht in gleicher oder in ähnlicher Form bei einer anderen Stelle zur Erlangung eines akademischen Grades eingereicht wurde.

**München, den 28.04.2015**

---

Ort, Datum



---

Unterschrift Doktorandin/Doktorand