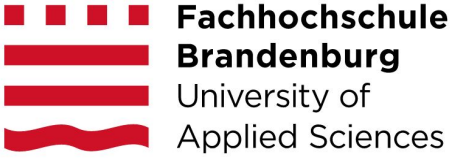

Fachbereich Informatik und Medien

BACHELORARBEIT

Definition und Implementierung einer CTS2-standardisierten Abbildung von Terminologien aus dem Bereich des Infektionsschutzes

Vorgelegt von: Franziska Krebs

am: 02.09.2013

zum
Erlangen des akademischen Grades

BACHELOR OF SCIENCE
(B.Sc.)

Erstbetreuer: Dipl.-Inf. Ingo Boersch

Zweitbetreuer: Dr. Andreas Billig

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit zum Thema

Definition und Implementierung einer CTS2-standardisierten Abbildung von Terminologien aus dem Bereich des Infektionsschutzes

vollkommen selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe. Die Arbeit wurde in dieser oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Brandenburg/Havel, den 02.09.2013

Unterschrift

Danksagung

Ich möchte mich recht herzlich bei meinen Betreuern Herrn Ingo Boersch und Herrn Andreas Billig für ihre Unterstützung bei der Anfertigung dieser Arbeit sowie ihre motivierenden Worte und Anregungen bedanken.

Weiterhin danke ich Herrn Kirchner vom Robert Koch-Institut für die Bereitstellung der verwendeten Daten.

Zusammenfassung

Der Einsatz von Begriffssystemen stellt in der heutigen Zeit keine Ausnahme mehr dar. Die rapide Entwicklung von einfachen Terminologien bis hin zu komplexen Ontologien wird angetrieben von dem Drang, das bestehende Wissen so zu formalisieren, dass es von einer Maschine verstanden und verarbeitet werden kann.

So zeigt diese Arbeit eine Implementierung, die einem modernen Ansatz der Terminologieverwaltung folgt. Von zentraler Bedeutung sind dabei die *Common Terminology Services (CTS2)*, welche eine Art Schablone für die Strukturierung von terminologischen Inhalten spezifizieren. Das Fraunhofer Institut für Offene Kommunikationssysteme bildete diesen Blueprint mithilfe des Resource Description Frameworks und des Schema-Vokabulars RDFS ab.
Umgesetzt wurde die Definition einer Abbildung von Terminologien aus dem Bereich des Infektionsschutzes sowie der Import dieser in einen RDF-Store. Die Ergebnisstruktur wurde anhand festgelegter Kompetenzfragen bewertet.
Die Evaluierung zeigte, dass alle Fragen vollständig und korrekt beantwortet werden konnten. Darüber hinaus war es möglich, ein Konzept in einem benachbarten Codesystem zu identifizieren und Wissen über dieses abzufragen.

Abstract

Nowadays, applying terminological content to numerous use cases is not an extraordinay thing to do anymore. The rapid development of simple code lists right up to full-fletched ontologies follows the needs to formalize knowledge in order to make it machine-readable. Adopting a modern approach in terminology administration, the Fraunhofer Institute developed a system, conformant to the *Common Terminology Services (CTS2)* standard. Because CTS2 only serves as a blueprint for building and accessing terminologies, an appropriate representaion format had to be chosen. For this purpose, the Fraunhofer Institute utilizes semantic technologies, such as the Resource Description Framework and the schema vocabulary RDFS.

The thesis demonstrates a mapping of code systems regarding infection protection to this specific implementation. Moreover, it comprises importing the terminologies into an RDF-store. Evaluation is done by checking if the code system is able to answer predefined competency questions. Not only did the terminology manage to answer the questions fully and correctly, it also illustrated cross-terminology querying based on an identified concept.

Inhaltsverzeichnis

1 Einleitung

Schon seit vielen Jahrhunderten sind die Menschen bemüht, ihr Wissen auf einem Fachgebiet mithilfe von Ordnungssystemen zu strukturieren. Bereits in der Antike ermöglichte in der Bibliothek von Alexandria die Katalogisierung einen Überblick über den Bestand der aufbewahrten Papyrusrollen. Das von Carl von Linné verfasste *Systema Naturea* dient noch fast 300 Jahre nach seiner Entstehung als Grundlage für biologische Nomenklaturen.

In der Moderne findet sich diese Wissensmodellierung wieder in vielfältigen Begriffssystemen, wie beispielsweise Glossaren und Thesauri. Dabei finden diese Konstrukte Einzug in verschiedene Disziplinen, in denen sie durch ihre Ordnung und Eindeutigkeit das Wissensmanagement nachhaltig bereichern. Insbesondere in der Informatik eröffneten sich zahlreiche Forschungsfelder, die sich der Entwicklung und Verwaltung komplexer Wissensnetze widmen.

1.1 Motivation

Motiviert wurde diese Bachelorarbeit einerseits durch ein großes Interesse an Techniken der Wissensrepräsentation und andererseits durch ein vom Institut bearbeitetes Projekt. Mit dem *Deutschen Elektronischen Meldesystem für Infektionsschutz (DEMIS)* soll eine medienbruchfreie Plattform geschaffen werden, die es ermöglicht, meldepflichtige Infektionskrankheiten zeitnah, unkompliziert und unter Einbezug der zuständigen Meldeempfänger zu kommunizieren. [Ben13, S. 20]

Der Einsatz standardisierter Codesysteme stellt besonders im Bereich der medizinischen Dokumentation ein bedeutendes Kriterium dar. Wenn sich alle Beteiligten auf die Verwendung eines gemeinsamen Vokabulars einigen, ist die Kommunikation medizinischer Termini eindeutig und Verständnisprobleme können vermieden werden. Zu diesem Zwecke entwickelte das Robert Koch-Institut Begriffssysteme für den Bereich des Infektionsschutzes, die im DEMIS-Projekt Verwendung finden sollen. Die Speicherung und Verarbeitung dieser Terminologien mittels semantischer Technologien entfernt sich von alltäglichen Datenbanksystemen und geht den Schritt hin zu komplexen, vernetzten Anwendungen, die sich Methoden der

Wissensrepräsentation wie Schlussfolgerungen und Anfragen via SPARQL bedienen. Durch die Orientierung an der CTS2-Spezifikation ergeben sich vielfältige Möglichkeiten in der Verteilung und Verwaltung von terminologischen Inhalten.

1.2 Aufgabenstellung

Im Zuge der Bachelorarbeit werden die Terminologien des Robert Koch-Instituts auf ihre Abbildbarkeit nach CTS2-Standard hin untersucht. Es erfolgt eine Anforderungsanalyse mit der Herausarbeitung möglicher Kompetenzfragen. Nach der Auswahl einer Untermenge werden die Export-Struktur und die Qualität der Daten untersucht. Anschließend wird die Abbildung der Codesysteme nach CTS2-Standard definiert und implementiert. Mithilfe der festgelegten Anforderungen wird die Ergebnisstruktur abschließend bewertet.

1.3 Abgrenzung

Die Arbeit umfasst keine auf den Projektrahmen von DEMIS zugeschnittene Implementierung der Abbildung. Vielmehr sollen vom Einsatz der Codesysteme im zukünftigen Meldesystem Anforderungen an die hier umgesetzte Abbildung abgeleitet werden.
Weiterhin wird in der Arbeit nicht die Implementierung von Schnittstellen zur Anbindung von Clients im Sinne der CTS2-Spezifikation vorgenommen.

1.4 Aufbau der Arbeit

In den ersten beiden Kapiteln werden die verwendeten Technologien und Werkzeuge vorgestellt sowie thematisch verwandte Arbeiten betrachtet. Das dritte Kapitel schildert das Konzept und die Umsetzung der Aufgabe. Anschließend wird eine Evaluation der Abbildung vorgenommen und die Ergebnisse werden zusammengefasst. Das Fazit beinhaltet einen Ausblick sowie ein Schlusswort.

2 Technologien und Werkzeuge

2.1 Wissenrepräsentation

Um Wissen gezielt verarbeiten zu können, muss zunächst eine geeignete Form einer maschinenlesbaren Modellierung gefunden werden. Die Lösung dazu findet sich im Einsatz *formaler Logiken*, welche die Grundlage für eine symbolische Repräsentation und zahlreiche Verarbeitungsalgorithmen bilden. Dies wird möglich durch eine genaue Definition der Syntax (Regeln für den Aufbau von Formeln) sowie der Semantik (der verwendeten Wahrheitsbegriffe).

In der Praxis ist die direkte Arbeit mit formalen Logiken ungeeignet. Man stelle sich vor, eine Anwendungsregel solle von einem Experten einer Domäne, etwa einem Mediziner, in ein bestehendes System integriert werden. Der Einarbeitungsaufwand hierfür wäre untragbar. Des Weiteren existieren zahlreiche syntaktische Strukturen, deren gleichzeitiger Einsatz zu fehlerhaften Inferenzen führen könnte. Zu diesem Zwecke wurden Repräsentationsformate entwickelt, die die Konstrukte einer formalen Logik auf vereinfachte Bausteine abbilden.

2.1.1 Repräsentationsformate

Es existieren verschiedene Repräsentationsformate, von denen 3. und 4. genauer erläutert werden:

1. Regeln (Formulierung von Wenn-dann-Beziehungen zur Repräsentation von Wissen)
2. Entscheidungsbäume (Darstellung von Wissen in einem gerichteten Baum)
3. Semantische Netze
4. Beschreibungslogiken

Die Idee, Sachverhalte mithilfe von netzartigen Gebilden darzustellen, wird bei semantischen Netzen aufgegriffen. Mind Maps und Ursache-Wirkungs-Diagramme sind Beispiele für Netze, die auch in komplexen Systemen einen Überblick ermöglichen.

Ein semantisches Netz wird in [SS08, S. 131] definiert als *"ein gerichteter Graph, bei dem jedem Knoten und jeder Kante eine Zeichenkette als Knoten- bzw Kantenbezeichner zugeordnet ist."*

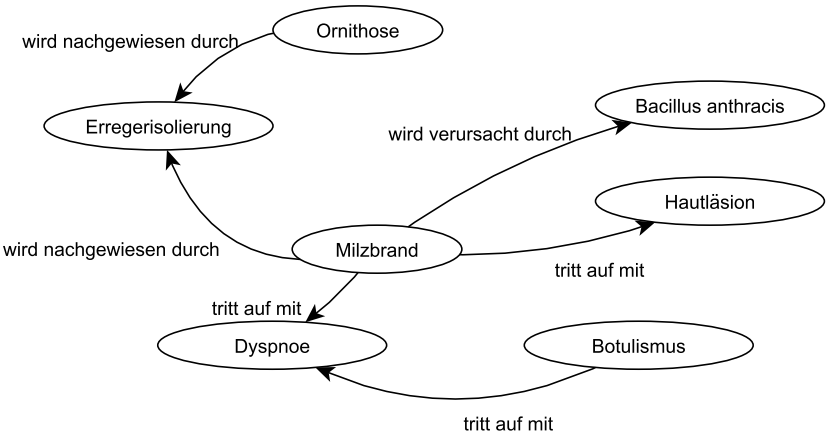

Abbildung 2.1: Beispiel für ein semantisches Netz

Ein Knoten entspricht dabei einem Konzept, eine Kante repräsentiert eine Beziehung zwischen diesen Konzepten. Die Abbildung 2.1 visualisiert ein semantisches Netz. Erkennbar sind verschiedene Begriffe, die über die Relationen "wird verursacht durch", "tritt auf mit" und "wird nachgewiesen durch" miteinander verknüpft sind. Die Bedeutung der Knoten kann schnell erschlossen werden: dargestellt werden die 3 Krankheiten *Ornithose*, *Milzbrand* und *Botulismus* sowie deren Symptome und Nachweismethoden.
Eine Abfrage von Wissen erfolgt in semantischen Netzen durch das Verfolgen von Knoten und Kanten. Da die Semantik nicht formal definiert ist, ist das Ziehen von Schlussfolgerungen nicht möglich. [SS08, S. 77,117 ff]

Die Beschreibungslogiken (auch *Description Logics*) gehen sowohl aus semantischen Netzen, als auch aus framebasierten Ansätzen der Wissensrepräsentation hervor. Dabei nutzen diese Logiken eine Untermenge der Prädikatenlogik, wodurch diese Sprachfamilie entscheidbar wird, d.h. für jede Fragestellung gibt es einen Algorithmus, der in endlicher Zeit eine Lösung findet. Dabei wird zwischen dem terminologischen Wissen (*T-Box*, Definition eines Vokabulars) und dem assertionalen Wissen (*A-Box*, Aussagen über Individuen und Terme des Vokabulars) unterschieden. Das Vokabular umfasst *Konzepte*, die Mengen von Individuen kennzeichnen, und *Rollen*, welche binäre Beziehungen zwischen den Individuen repräsentieren. [BCM+10, S. 50 f]

2.1.2 Terminologien

Als Terminologie bezeichnet man grundlegend die Menge aller Begriffe und ihrer Benennungen eines Fachgebietes. Es existieren unterschiedliche Strukturen zur Anordnung dieser Menge. Das Spektrum reicht hier von einfachen Katalogen über Glossare bis hin zu Thesauri.
In vielen Bereichen ist neben der Festlegung eines gemeinsamen Vokabulars ebenfalls die Eindeutigkeit eines Begriffes von zentraler Bedeutung. Ein kontrolliertes Vokabular vereinigt diese Aspekte, indem Homonyme vermieden werden.
Eine besondere Stellung unter den Terminologien nehmen Systeme ein, welche Begriffe durch Beziehungen so miteinander verbinden, dass die Semantik effizienter erschlossen werden kann.
[Hah06]

2.1.3 Semantic Web

> *"People can't share knowledge if they don't speak a common language "*
>
> Thomas Davenport

Das *Semantic Web* stellt eine von Tim Berners-Lee vorgeschlagene Lösung zur Weiterentwicklung des World Wide Webs dar. Dabei sollen Maschinen in der Lage sein, von Menschen getroffene Aussagen zu verarbeiten. Zu diesem Zweck werden semantische Netze in Verbindung mit mächtigen Werkzeugen der Künstlichen Intelligenz eingesetzt. Diese Kombination ermöglicht nicht nur eine Erfassung der Semantik auf formaler Ebene, sondern auch das Ziehen von Schlussfolgerungen.[JFH03, S. 33–40]
Eine wichtige Rolle bei der Bewältigung dieser Aufgabe spielen die Ontologien, die von T.R. Gruber als "eine formale, explizite Spezifikation einer gemeinsamen Konzeptualisierung" definiert werden. [Gru93] Gemeint ist damit eine von Maschinen interpretierbare Beschreibung eines Ausschnittes der realen Welt durch Konzepte und deren Beziehungen.
Der Aspekt der Formalität wird erreicht durch verschiedene Sprachen, die sich in ihrer Ausdruckskraft unterscheiden.

2.1.3.1 Resource Description Framework und RDF-Schema

Das Resource Description Framework (RDF) beschreibt Ausschnitte der realen Welt, indem Aussagen über Resourcen getätigt werden. Die Aussagen entsprechen sogenannten *Tripeln*,

also Ausdrücken der Form *Subject-Predicate-Object*. Die Ausdruckskraft von RDF beschränkt sich auf die Beschreibung von assertionalem Wissen. So können

- Relationen zwischen Instanzen und
- Klassenzugehörigkeiten mithilfe von in der T-Box definierten Klassen

ausgedrückt werden.

Für einfache Konstrukte in der T-Box eignet sich die Verwendung des *Resource Description Framework Schemas*. Es erlaubt die Definition von Klassen, Klassenhierarchien, Relationen (in RDF-Schema sog. *Properties*) und Relationshierarchien. Neben *Labeln*, die eine für den Menschen besser lesbare Bezeichnung einer Resource zulassen, erweitern Kommentare und Wertebereiche einer Relation (im Sinne von Klassenrestriktionen) das Vokabular des RDF-Schemas.

Mächtigere Ontologie-Beschreibungssprachen bauen auf den obig beschriebenen Formaten auf und erweitern diese durch den Einsatz von Beschreibungslogiken zur Formalisierung der Semantiken. Die *Web Ontology Language (OWL)* gilt als Nachfolger der framebasierten Sprache *DAML+OIL* und steigert die Ausdruckskraft durch umfassende Möglichkeiten zur Beschreibung von Rollen. [Fen04, S. 43-46]

Zur Speicherung von Daten in diesen Formaten existieren sogenannte Triple-Stores. Als Abfragesprache entwickelte das World Wide Web Consortium die graphbasierte **SPARQL Protocol And RDF Query Language**.

2.1.3.2 Jena-Framework

Das von Apache bereitgestellte Java Framework ermöglicht eine Entwicklung von Semantic Web Anwendungen. Das Jena-Projekt umfasst die folgenden Funktionen:

- eine API zum Lesen, Verarbeiten und Schreiben von RDF-Datensätzen in den Formaten XML, N-Tripeln und Turtle
- eine Ontologie API zum Verarbeiten von OWL und RDFS-Ontologien
- eine regelbasierte Inferenzmaschine zum Schlussfolgern aus RDF und OWL-Datenquellen
- eine Query-Engine für SPARQL

Die persistente Speicherung der Graphen sowie die Abfrage mittels SPARQL kann mithilfe der Jena-TDB Komponente in einem RDF-Triple Store erfolgen. [The]

2.2 Common Terminology Services Release 2

Bei den Common Terminology Services Release 2 (CTS2) handelt es sich um eine von den Organisationen *OMG* und *HL7* entwickelte Spezifikation zur Repräsentation von terminologischen Inhalten. Diese Inhalte variieren von einfachen Code- oder Termlisten bis hin zu komplexen Ontologien. CTS2 setzt dabei die folgenden Aspekte um:

- Schaffung einer standardisierten Schnittstellen-Spezifikation für den Zugriff auf eine Terminologie
- Identifizierung von funktionalen Anforderungen an einen Terminologie-Dienst
- Trennung von Inhalt und Funktionalität einer Terminologie
- Anbieten eines gemeinsamen Eintrittspunktes für den Zugriff und die Verwaltung einer Terminologie

[Sta11]

2.2.1 Komponenten der Spezifikation

Die Spezifikation ist unterteilt in 2 Komponenten, die im Folgenden erläutert werden.

Das **Platform Independent Model** wird genutzt, um Plattform-unabhängige Eigenschaften zu definieren, wohin gegen das **Platform Specific Model** Plattform-spezifische Informationen einer im System umgesetzten Technologie beinhaltet.

In beiden Komponenten wird eine weitere Unterteilung vorgenommen:

- Information Model
  Dieses stellt eine statische Spezifikation von in strukturierten Terminologien gemeinsam vorkommenden Elementen zur Verfügung. Mithilfe von Klassendiagrammen werden die jeweiligen Attribute und Relationen zu anderen Resourcen veranschaulicht.
- Computational Model
  Das Computational Model spezifiziert Methoden und Schnittstellen für den Zugriff und die Verwaltung der Elemente.

Grundsätzlich ist diese Spezifikation nicht als lauffähiges System zu betrachten. Vielmehr wurde hier ein gemeinsames strukturelles Model projektiert, welches unabhängig vom Inhalt oder dem Verhalten einer Terminologie in einer spezifischen Systemumgebung implementiert werden kann.

Implementierung

Bei der Entwicklung einer CTS2-konformen Anwendung gilt das Prinzip der Modularität: es wird nur die Funktionalität implementiert, die für einen bestimmten Anwendungsfall benötigt wird. [Obj11a]
Zu diesem Zwecke existieren verschiedene Profile für die Implementierung der Resourcen und der Dienste. Soll eine Terminologie beispielsweise lediglich einsehbar und abfragbar sein, so genügt es, die Dienst-Profile “Read“ und “Query“ umzusetzen. Ein Überblick über weitere Profile wird in Anhang A.1 gegeben.

Weiterhin ist eine geeignete Datenstruktur für die Abbildung des Information Models auszuwählen. CTS2 legt keine Restriktionen hinsichtlich des Datenformates fest und somit können beliebige Repräsentationen (XML-, SQL-Datenbanken, RDF-Triple-Stores) eingesetzt werden. [Sta11, S. 42]

Das Fraunhofer FOKUS entwickelte ein CTS2-konformes System mit dem Namen **CTS2-Le**, welches im folgenden Abschnitt vorgestellt wird.

2.2.2 CTS2-Le

CTS2-Le ist ein System, welches dem CTS2-Standard entsprechend entwickelt wurde. Dabei wurde

1. eine Abbildung eines veränderten Information Models vorgenommen, wobei als
2. Repräsentationsformat das Resource Description Framework Schema

ausgewählt wurde.

Systemarchitektur

Die Abbildung 2.2 zeigt die grobe Architektur des in Java programmierten CTS2-Le-Systems.

Die Persistenzschicht setzt sich zusammen aus einem RDF-Quad-Store (*vom Jena-Framework bereitgestellter TDB-Store*) und einer Volltextindizierungs-Komponente (Apache Lucene).
In der darüber liegenden Schicht erfolgt die Erstellung der konkreten Wissensnetze mithilfe des Jena-Frameworks. Außerdem bieten die sog. *Negros-Utilities* einen Schema- sowie Query-Validator innerhalb einer leichtgewichtigen *SPARQL-Engine* an. Zur Validierung eines Codesystems gegen die CTS2-Spezifikation werden RDF-Signaturen, welche als Schema-Definition fungieren, verwendet. Bereits vordefinierte *SPARQL-Templates* erleichtern einen Zugriff auf die Terminologien.

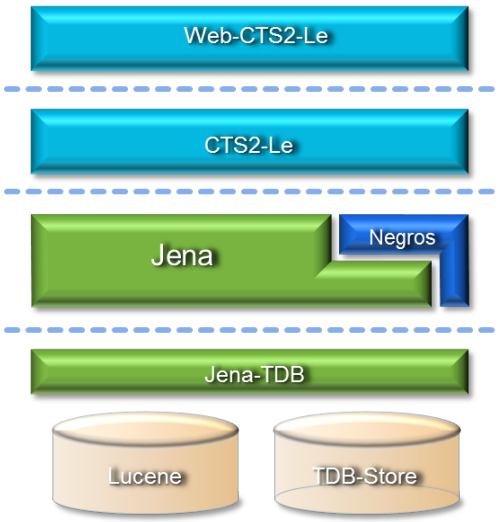

Abbildung 2.2: CTS2-Le Systemarchitektur [1]

Die beiden oberen Schichten(*Web-CTS2-Le* und *CTS2-Le*) beinhalten die folgenden Komponenten:

- konkrete Java-Klassen, welche eine Abbildung des jeweiligen Codesystems implementieren
- Anbindung möglicher CTS2-Clients über die Service-Schnittstellen
- CTS2-Navigator[2] für explorative Zwecke und Suchfunktionen

[Bil13]

zu 1.) Vorstellung des veränderten Information Models

In diesem Abschnitt werden die wesentlichen Elemente des Models erläutert. Ebenfalls beschrieben wird die Art und Weise der Repräsentation in RDF-Schema.
Im Anhang A.2 kann das vollständige Information Model eingesehen werden. Die Abbildung A.2 und die folgenden Ausschnitte sind UML-ähnliche Modelle, welche aus den RDF-Signaturen generiert wurden. Darüber hinaus bestimmen diese (an Frame Logic [KLW95] orientierten) Signaturen maßgeblich die unter Punkt 2 beschriebene Repräsentation nach RDF-Schema.

Die Spezifikation lässt sich - je nach beschriebener Domäne - unterteilen in verschiedene Abschnitte. Neben Bereichen, die sich der Erläuterung von Code Systemen, Code System

[1] Grafik verändert entnommen aus [Bil13]
[2] erreichbar unter `http://semantik.fokus.fraunhofer.de/WebCts2LE/main3/ini.jsp`, letzter Zugriff am 05.08.13
Zum jetzigen Zeitpunkt werden ausschließlich Standardbeziehungen (Hierarchien) im Browser präsentiert.

Versionen und Value Sets widmen, sind vor allem die Abschnitte **Entity Description** und **Association** von besonders hoher Relevanz.
Ersteres befasst sich mit der Beschreibung von Entitäten (im Sinne von “Konzepten“, “Klassen“, “Prädikaten“, “Relationen“, “Termen“ oder “Individuen“) und stellt dafür Werkzeuge wie *Designations* oder *Definitions* bereit. Der Bereich Association ermöglicht das Treffen von “semantischen Aussagen“ über eine Relation zwischen Klassen bzw. über eine Relation zwischen einer Klasse und einem Literal. [Obj11b]

Die Entitäten, die diese Aufgaben erfüllen, sind in dem folgenden Ausschnitt 2.3 des Information Models zu erkennen und werden in der Tabelle im Anhang A.2 näher erläutert. Dabei wird zur Angabe der Kardinalitäten die Erweiterte Backus-Naur-Form (EBNF) mit folgender Notation verwendet:

- ? entspricht 0..1
- * entspricht 0..*
- + entspricht 1..*

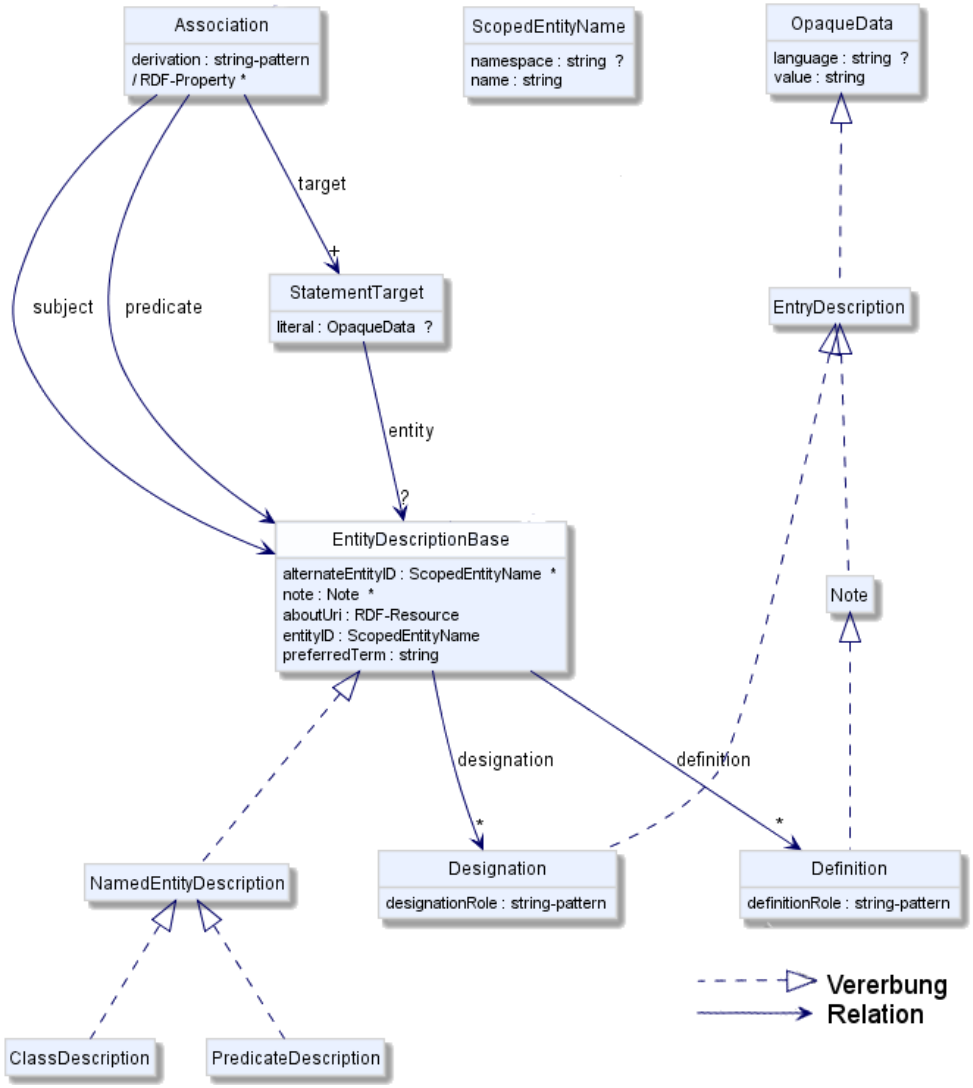

Abbildung 2.3: Entitäten der Bereiche Entity Description und Association

zu 2.) RDF und RDF-Schema als Repräsentationsformate des Information Models

Die Abbildung erfolgt mithilfe von RDF, RDFS sowie eines eigenen Schema-Vokabulars[3] und ist wie folgt definiert:

- Eine Entität im CTS2-Model entspricht einer rdfs:Class.
- Eine Vererbungsbeziehung im CTS2-Model entspricht einer rdfs:subClassOf-Property.
- Attribute einer Entität entsprechen rdf:Properties, wobei der Attributname dem Namen der Property entspricht und der Datentyp des Attributes durch sig:range als Datentyp des Objektes der Property festgelegt wird. Weiterhin wird die Entität des Attributes als sig:domain der Property definiert.

[3]Das eigene Schema-Vokabular ist erkennbar an dem Prefix **sig**.

- Kanten zwischen Entitäten entsprechen rdf:Properties, wobei der Kantenname dem Namen der Property entspricht und die Entität am Pfeilende durch sig:range als Datentyp des Objekts der Property festgelegt wird. Weiterhin wird die Entität am Pfeilanfang als sig:domain der Property definiert.
- Kardinalitäten einer Beziehung bzw. eines Attributes werden jeweils durch sig:min, sig:max und Kombinationen dieser Resourcen definiert.

Beispielsweise wird die Entität *EntityDescriptionBase* abgebildet als rdfs:Class, wobei das Attribut *preferredTerm* einer rdf:Property entspricht. Ebenso werden die Entitäten *NamedEntityDescription* und *Designation* als rdfs:Class definiert. Die Kante *designation* wird repräsentiert durch eine rdf:Property.
Unter Einbezug der Hierarchierelationen und Domain- und Range-Angaben ergibt sich die folgende RDF-Syntax[4]

```
<rdfs:Class rdf:ID="EntityDescriptionBase"/>
<rdfs:Class rdf:ID="NamedEntityDescription">
  <rdfs:subClassOf rdf:resource="EntityDescriptionBase"/>
</rdfs:Class>

<rdfs:Class rdf:ID="Designation"/>

<rdf:Property rdf:ID="preferredTerm">
  <sig:domain rdf:resource="EntityDescriptionBase"/>
  <sig:range rdf:resource="xsd:string"/>
</rdf:Property>

<rdf:Property rdf:ID="designation">
  <sig:domain rdf:resource="EntityDescriptionBase"/>
  <sig:range rdf:resource="Designation"/>
</rdf:Property>
```

Listing 2.1: Repräsentation der CTS2-Entitäten in RDFS

Den hier beschriebenen Graphen zeigt die Abbildung 2.4.

[4]Es wird darauf hingewiesen, dass der RDF-Auszug lediglich die im Text beschriebene Abbildung definiert. Die vollständige Abbildung der erläuterten Entitäten kann anhand der im Anhang Listing A.1 dargestellten Signaturen eingesehen werden.

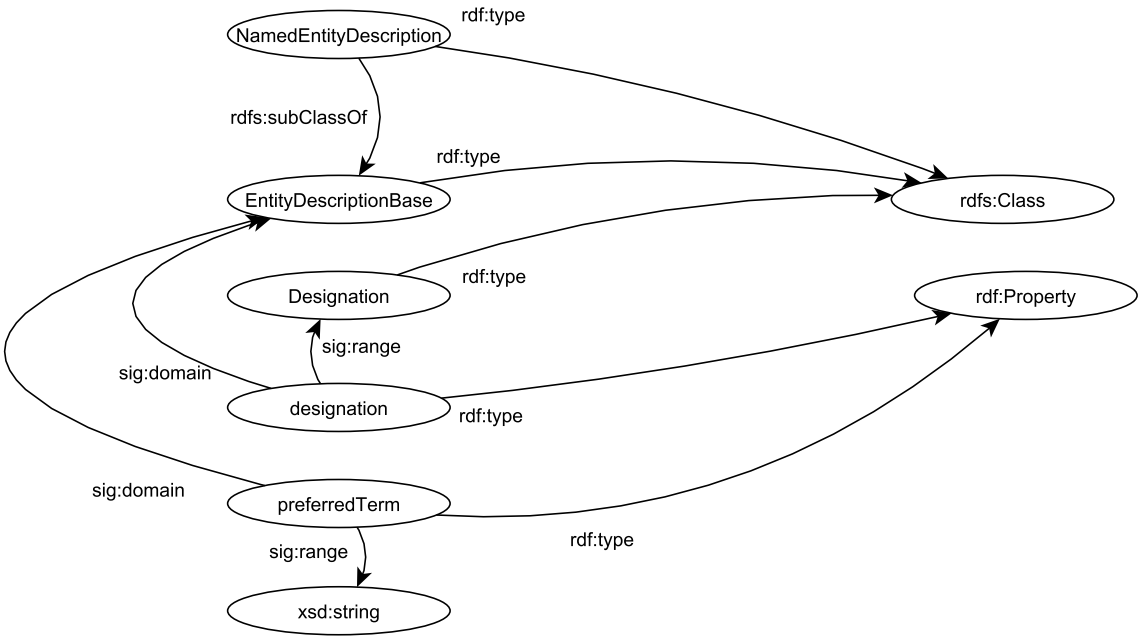

Abbildung 2.4: Darstellung des Graphen

3 Verwandte Arbeiten

3.1 LexGrid und LexEVS

Die Organisation *Mayo Clinic* entwickelte in Zusammenarbeit mit dem *National Cancer Institute* zahlreiche Komponenten, deren Funktionen sich mit denen der CTS2-Spezifikation überschneiden.
In Anlehnung an das Information Model der CTS2-Spezifikation wurde ein gemeinsames terminologisches Model (LexGrid) entwickelt, welches einen offenen Zugang zu Terminologien und Value Sets bietet sowie ein Cross-Terminologie-Matching umsetzt. Das LexGrid (Lexical Grid)-Model definiert weiterhin verschiedene Mechanismen der Datenhaltung und ermöglicht die Repräsentation von verschiedenen Quellformaten. So wird beispielsweise das Vokabular des NCI Thesaurus (eine umfassende Terminologiedatenbank) mithilfe der Web Ontology Language repräsentiert. Die Common Terminology Services und die LexBIG API[1] sind 2 Beispiellösungen für den Zugriff auf das Model.
Eine Kombination der LexBIG API und der Enterprise Vocabulary Services (eine Sammlung von Werkzeugen und Diensten zur Erleichterung der Terminologiearbeit im biomedizinischen Sektor) führte zur Entwicklung des Terminologieservers LexEVS.
[Nat12]

3.2 BioPortal

Das *NCBO Bioportal*[2] stellt einen webbasierten Zugriff auf Ontologien im medizinischen Bereich dar. Zunächst wurde dieser mittels einer BioPortal REST API realisiert. Da zu diesem Zeitpunkt die Entwicklung der CTS2-Spezifikation absehbar war, sollte diese ebenfalls als Zugriffsmöglichkeit bereitgestellt werden.
Im Back-End findet sich der LexEVS Terminologie-Server als Komponente wieder. [Nat]

[1] eine im Rahmen der Cancer Biomedical Informatics Grid-Initiative entwickelte API
[2] `http://bioportal.bioontology.org/`, letzter Zugriff am 14.08.13

4 Definition und Implementierung einer CTS2-standardisierten Abbildung

4.1 Konzeption

Der erste Schritt bestand in der Analyse der Terminologien, welche vom Robert Koch-Institut entwickelt wurden. Als Untersuchungskriterien wurden der Inhalt und der Aufbau der Vokabularien festgelegt, mit dem Ziel, Konzepte und Konzeptstrukturen für die Abbildung zu identifizieren. Anhand dessen soll festgestellt werden, welche Elemente für die Abbildung nach CTS2-Standard geeignet sind. Angelehnt an das methodische Vorgehen bei der Entwicklung von Ontologien gemäß *Ontology Development 101* [MN01], erfolgt die Erfassung der Anforderungen mit der Herausarbeitung von vorstellbaren Kompetenzfragen. Diese werden ebenso wie die Ergebnisse der Repräsentierbarkeitsanalyse für die Auswahl einer aussagekräftigen, evaluierbaren Untermenge herangezogen. Anschließend erfolgt die Definition und Implementierung der Abbildung. Anhand der erarbeiteten Anforderungen wird die Ergebnisstruktur bewertet.

4.2 Analyse der Terminologien

Im folgenden Abschnitt werden die inhaltlichen und strukturellen Aspekte der Terminologien[1] vorgestellt. Im Wesentlichen sind dies

1. Krankheiten
2. Krankheitsformen
3. Erreger
4. Nachweismethoden
5. Symptome und
6. Impfstoffe.

1.) Krankheiten
Kommt es zur Erkrankung einer Person an einer meldepflichtigen Krankheit oder besteht der Verdacht einer Infektion, so entspricht dies einem Meldetatbestand gemäß §7(1) Infektionsschutzgesetz. Dieser ist nach gesetzlichen Regelungen an eine Institution des Gesundheitswesens zu übermitteln.
Das Begriffssystem beschreibt 75 der Meldepflicht unterliegenden Krankheiten durch die folgenden Attribute:

- Angabe der Bezeichnung des Erregers
- Angabe des Paragraphen oder der Verordnung eines Bundeslandes, nach dem eine Krankheit meldepflichtig ist
- Angabe des Bundeslandes, in dem eine Krankheit meldepflichtig ist
- Angabe des Codes im ICD-10-Codesystem (wenn vorhanden)
- Angabe, ob eine Impfung möglich ist

Zwischen den Krankheiten bestehen keine Hierarchie-Relationen, jedoch existieren Beziehungen zu den nachfolgend erläuterten Konzepten.

2.) Krankheitsformen
Vereinzelt werden für eine Krankheit verschiedene Formen definiert. Diese werden in einer Liste gruppiert, deren Bezeichnung sich aus "Form" und dem Krankheitskürzel zusammensetzt. Die Anordnung der Krankheitsausprägungen ist nicht hierarchisch.

[1] Im Folgenden werden die Wörter "Terminologie", "Begriffssystem" und "Codesystem" als Synonyme verwendet.

3.) Erreger

Diese Terminologie definiert für jede meldepflichtige Krankheit eine formale Liste, welche durch eine Menge von Erregern beschrieben wird. Darüber hinaus werden Listen vereinzelt für die Definition von spezifischeren Formen eines Erregers (z.B. Serotypen[2] und Lysotypen[3]) herangezogen.

Beispiel

Der Krankheit "Adenovirus-K(eratok)onjunktivitis" mit dem Kürzel "ADV" wird eine Liste mit dem Namen «PathogenADV» zugewiesen. Die Listenelemente entsprechen den eigentlichen Erregern. Die Abbildung A.1 im Anhang zeigt die verschiedene Typen des Virus.

In jeder Erreger-Liste sind weiterhin die für eine Meldung benötigten Felder "nicht ermittelbar", "nicht erhoben" und "andere/sonstige" enthalten.

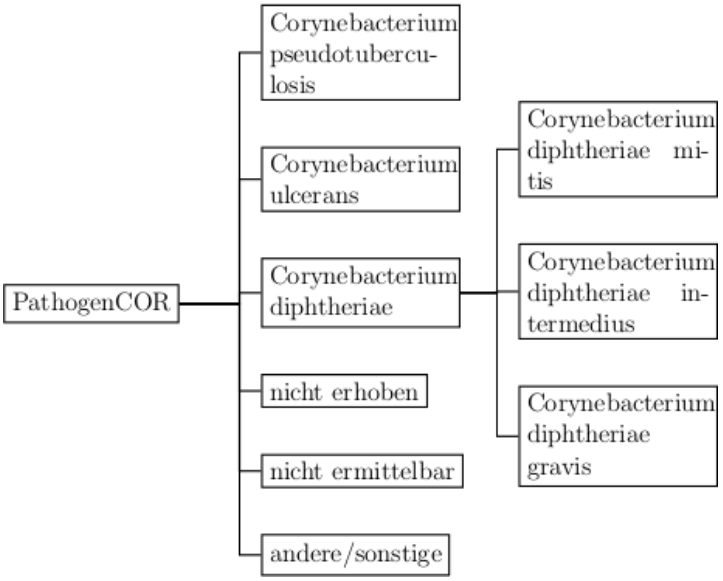

Abbildung 4.1: Hierarchie der konkreten Erreger der Diphtherie

Wie die Abbildung 4.1 am Beispiel der Diphtherie visualisiert, können Hierarchie-Relationen zwischen Elementen der Listen existieren. Eine Subsumtion auf Listen-Ebene wird nur im Falle der Lysotyp- oder Serotyp-Spezifizierung vorgenommen.

4.) Nachweismethoden

Die Terminologie der diagnostischen Maßnahmen zum Nachweis einer meldepflichtigen Krankheit zeichnet sich aus durch wenige Hierarchien. Während der Großteil der Methoden unstrukturiert vorliegt, haben sich vereinzelt Konzepte spezialisiert. Das Hypernym "Nukleinsäure-Nachweis (z.B. PCR) aus Sekret des Respirationstraktes" wird beispielsweise

[2] eine subspezifische Einteilung eines Erregers aufgrund der sich auf der Oberfläche befindlichen Antigene
[3] ein eindeutig abgegrenzter Bakterienstamm

durch die Angabe des untersuchten Sekretes (“aus Sputum“, “aus Trachealsekret“, “aus broncheoalveolärer Lavage (BAL)“) ergänzt. Weiterhin werden für 2 Krankheiten Methodenlisten definiert, die dem Schema der vorherig beschriebenen Listen folgen.

5.) Symptome

Die Auswertung der Krankheitserscheinungen ergab ähnliche Ergebnisse wie die der Methoden. Ein Teil der ca. 300 Symptome wurde generalisiert und entsprechend spezifischeren Konzepten übergeordnet. Zum Beispiel verallgemeinert das Konzept “Schmerzen“ die konkreten Begriffe “Rückenschmerzen“, “ Kopfschmerzen“ und “Muskelschmerzen“.

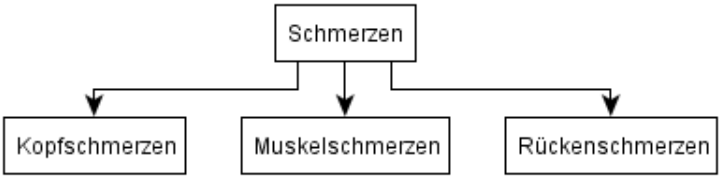

Abbildung 4.2: Beispiel einer Hierarchie in der Terminologie der Symptome

6.) Impfstoffe

Sofern für eine Krankheit eine präventive Maßnahme in Form einer Schutzimpfung durchgeführt werden kann, steht für diese jeweils ein nicht-hierarchischer Katalog von Impfstoffen zur Verfügung.

Tabelle 4.1: Impfstoffliste der Diphtherie

Krankheit	Name der Impfstoffliste	Elemente der Liste
Diphtherie	«VaccineCOR»	Diphtherie-Kombinationsimpfstoffe mit Pertussiskomponente (D/d-Kombination mit aP/P)
		Diphtherie-Kombinationsimpfstoffe ohne Pertussiskomponente (D/d-Kombination ohne aP/P)
		monovalenter Diphtherie-Impfstoff (D/d)
		nicht ermittelbar
		nicht erhoben

Der Name einer Impfstoffliste setzt sich zusammen aus “Vaccine“ und dem jeweiligen Krankheitskürzel. Die in Tabelle 4.1 beschriebene Liste trägt den Namen «VaccineCOR» und zählt 3 Impfstoffe der Krankheit Diphtherie auf. Hinzu kommen die für eine Meldung relevanten Felder “nicht ermittelbar“ sowie “nicht erhoben“.

Die dargestellten Codesysteme setzen eine konkrete Abbildung der Domäne um und strukturieren das Vokabular durch Generalisierungsbeziehungen.

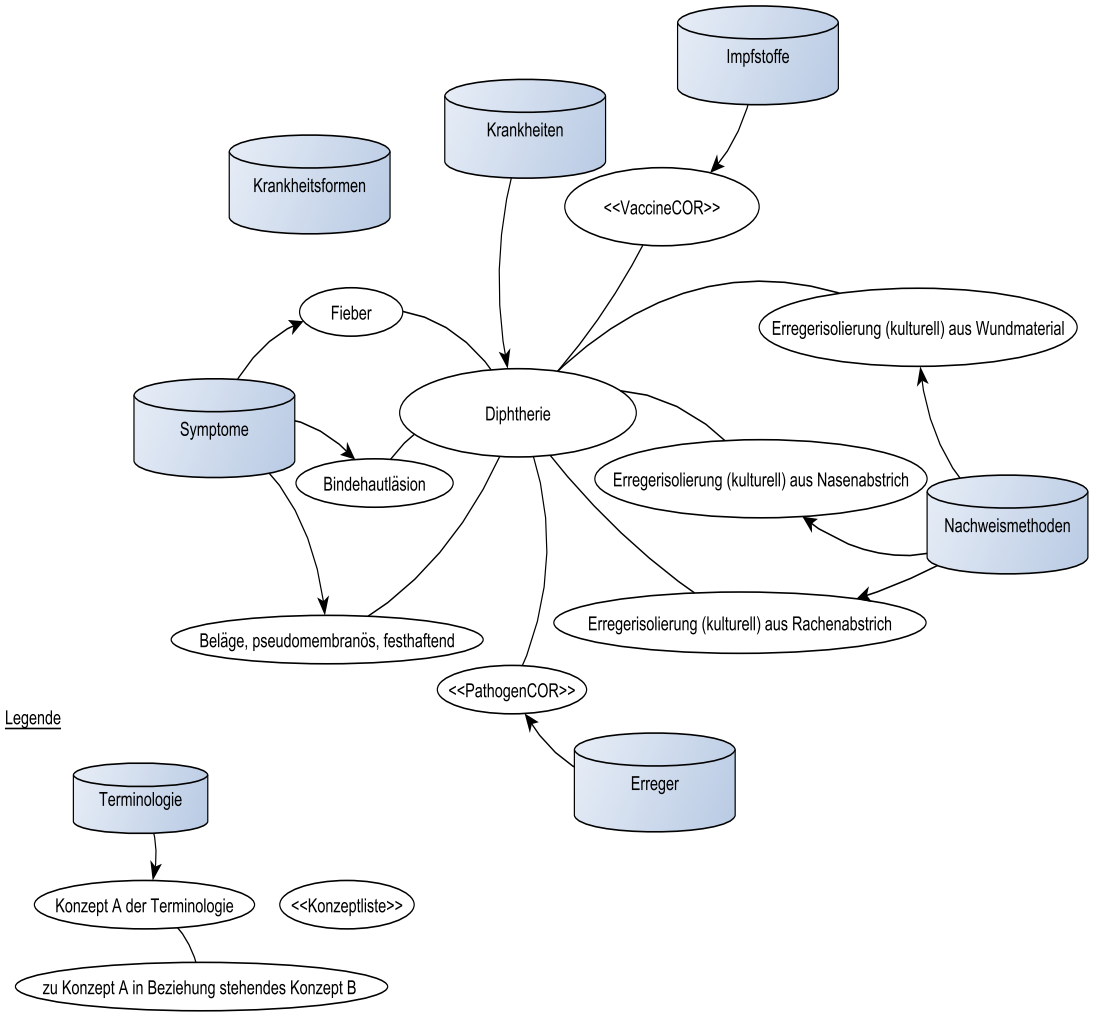

Abbildung 4.3: Demonstration des Zusammenspiels der Terminologien am Beispiel der Diphtherie

Durch die Vernetzung des zentralen Konzeptes der Krankheiten mit Value Sets anderer Terminologien entsteht ein ganzheitliches Begriffssystem, welches die Abbildung 4.3 auszugsweise präsentiert. Im Zentrum des Netzes befindet sich die Krankheit “Diphtherie“, die beschrieben wird durch ihre Relationen zu Erregern, Methoden, Krankheitserscheinungen und Impfstoffen. Da diese Erkrankung isoliert dargestellt wird, sind gemeinsame Konzepte, beispielsweise das Auftreten des Symptoms “Fieber“ bei mehreren Erkrankungen, außer Acht gelassen.

4.3 Anforderungsanalyse

Die Anforderungsanalyse orientiert sich am *Ontology Development 101*, welches in 7 Schritten eine iterative Vorgehensweise zur effektiven Modellierung von Wissen präsentiert. Der Prozess setzt sich zusammen aus den Phasen:

1. Determine Scope
2. Consider Reuse
3. Enumerate Terms
4. Define Classes
5. Define Properties
6. Define Constraints
7. Create Instances

[MN01]
Im Rahmen der hier durchgeführten Analyse wird lediglich der erste Punkt eingehend bearbeitet.

Geltungsbereich und Ziel
Die Terminologien decken die Domäne der in Deutschland meldepflichtigen Infektionskrankheiten ab und präsentieren die grundlegenden Konzepte zur Beschreibung einer solchen Krankheit. Dabei handelt es sich vielmehr um einen Vorschlag der effizienten Strukturierung dieser Begriffssysteme mithilfe der erläuterten CTS2-Spezifikation, als um eine Neuentwicklung.
In diesem Sinne soll eine Erweiterung der im CTS2-Le-System bestehenden Terminologiemenge erreicht, als auch eine Anbindung an diese umgesetzt werden. Ein vorstellbares Anwendungsszenario wäre zunächst das Erforschen der Terminologieinhalte mittels des Terminologiebrowsers. Da eine vollständige Abbildung im Rahmen des DEMIS-Projektes erfolgen soll, ergeben sich neue Aufgabenfelder. Beispielsweise assistiert das System bei der Erfassung von Meldungen und nimmt eine unterstützende Funktion ein, indem der Melder an die Abfrage relevanter Informationen erinnert wird. Aus diesen Anforderungen lassen sich Fragen skizzieren, die die Terminologien beantworten sollen:

Herausarbeitung von Kompetenzfragen (Competecy Questions)

CQ1 Welche meldepflichtigen Krankheiten gibt es?

CQ2 Welche Krankheiten sind im Bundesland Nordrhein-Westfalen meldepflichtig?

CQ3 Welche Symptome treten bei der Campylobacter-Infektion auf?

CQ4 Welche Methoden zum Nachweis der Ornithose gibt es?

CQ5 Kann der Krankheit Diphtherie durch eine Impfung entgegen gewirkt werden? Wenn ja, welche Impfstoffe existieren?

CQ6 Welche Erreger führen zu einer Krankheit mit dem "Symptom Dyspnoe (Atemstörung)"?

CQ7 Ist Fieber ein Symptom der EHEC-Erkrankung?

CQ8 Welche Krankheitsformen des Milzbrandes gibt es?

Die Stärke der Modellierung mithilfe der CTS2-Spezifikation liegt vor allem in der Möglichkeit der Vernetzung mit eigens implementierten Terminologien. Derzeitig umgesetzte Codesysteme, deren Nutzen in Betracht gezogen werden sollte, sind der *OPS* und das *ICD*-Codesystem. Da die Terminologien des Robert Koch-Instituts zu einer Krankheit explizit den ICD-Code bereitstellen, erscheint der Einsatz dieses Codesystems aufgrund der eindeutigen Zuordnung sehr sinnvoll.
Anhand der Definition einer Nachweismethode lässt sich hingegen nicht ableiten, welche Prozedur im OPS gemeint ist. Hier liegt auch die Vermutung nahe, dass bei der Entwicklung der Terminologien eine Einordnung der Methoden in den OPS nicht angestrebt wurde. Weitere Kompetenzfragen beschränken sich demnach auf die Verwendung des ICD-Codesystems und lauten wie folgt:

CQ9 Unter welcher Kategorie ist die Krankheit Botulismus im ICD-10 eingeordnet?

CQ10 Welche weiteren Ausprägungen sind im ICD-10 Codesystem für das Fleckfieber definiert?

CQ11 Gehören "sonstige oberflächliche Mykosen" (Codierung im ICD-10: B36) zu meldepflichtigen Krankheiten?

Angezielt wird die Beantwortung aller Kompetenzfragen durch die Terminologien.

4.4 Analyse der Export-Struktur

Im folgenden Abschnitt wird das zum Export genutzte System *SurvNet@RKI* [Rob13] vorgestellt. Die daraus gewonnene Datei stellt die Arbeitsgrundlage für die weiteren Schritte dar und wird zu diesem Zwecke strukturell untersucht. Der Fokus der Analyse liegt dabei auf der Repräsentation der im Abschnitt 4.2 erläuterten Terminologien.

SurvNet@RKI

Das Robert Koch-Institut entwickelte zum Zwecke der Erfassung, Weiterleitung und Auswertung von Meldetatbeständen gemäß Infektionsschutzgesetz das System SurvNet@RKI. Grundsätzliches Ziel der aktuell in der 3. Version erhältlichen Software ist die Veröffentlichung von Datenanalysen, die auf der Basis der gemeldeten Fälle durchgeführt wurden. Selbstständige Abfragen dieser Publikationen können über die Web-Schnittstelle *SurvStat@RKI*[4] getätigt werden.

SurvNet@RKI erlaubt den Export der Metadaten in die hier genutzte XML-Datei. Die Terminologien sind dabei eingebettet in Datenstrukturen, die in erster Linie für die Erstellung einer Meldung und den Transport ebenjener konzipiert wurden. Da diese nicht Gegenstand des festgelegten Scopes sind, wird auf die Erläuterung dieser Inhalte verzichtet.

Vorstellung der XML-Struktur

Die Strukturen der XML-Datei setzen die im Anhang B beschriebenen Entitäten des Datenbankschemas um. Dieser Abschnitt wurde angereichert mit kurzen Ausschnitten aus der XML-Datei, um die wesentlichen Elemente

- **Diseases** zur Beschreibung der Krankheiten,
- **Fields** zur Beschreibung von Symptomen und Methoden,
- **Catalogues** zur Beschreibung von Erreger-, Impfstoff-, Methoden- und Krankheitsformlisten,
- **MigrationSurvNet3s** zur Zuordnung der Felder und Kataloge zu Krankheiten

zu veranschaulichen.

Die Analyse brachte die folgenden Ergebnisse hervor:

1. Eine Krankheit wird jeweils durch ein Child-Element “Disease“ definiert. Die Meldepflicht einer Krankheit gemäß einer Länderverordnung und die Angabe der Bundesländer, in denen eine Krankheit meldepflichtig ist, werden durch die Attribute

[4] `http://www3.rki.de/SurvStat/`, letzter Zugriff am 21.08.13

IfSGBundesland und *InBundesland* festgelegt. Ersteres kann die Werte 0 (nicht meldepflichtig) und 1 (meldepflichtig) annehmen, Bundesländer werden durch die Ziffern 01 bis 16 durch Kommata getrennt repräsentiert.

```
<Diseases>
  <Disease Code="ADV" DiseaseName="Adenovirus-K(eratok)
     onjunktivitis" SpecimenName="Adenovirus im
     Konjunktivialabstrich" IfSGBundesland="0" InBundesland=
     "01,02,03,04,05,06,07,08,09,10,11,12,13,14,14,16"
     ICD10Code="B30.0">
</Diseases>
```

Listing 4.1: in XML notierte Krankheit Adenovirus-K(eratok)onjunktivitis

Durch die Angabe des Kürzels (*Code*), der Krankheitsbezeichnung (*DiseaseName*), der Erregerbezeichnung (*SpecimenName*) und des Codes im ICD-10 System (*ICD10Code*) wird die Beschreibung komplettiert.

2. Ein Feld wird jeweils durch ein Child-Element “Field“ definiert. Zur eindeutigen Identifizierung eines Feldes wird eine “IdField“ festgelegt. Ein “FieldName“ stellt eine Art Kodierung dar, anhand derer abgeleitet werden kann, ob dieses Feld ein Symptom oder eine Methode repräsentiert. Die Zurückführung ist sehr intuitiv, da der “FieldName“ sich entsprechend aus “Symptom“ bzw. “Method“ und einer Zahlenfolge zusammensetzt (siehe Listing 4.2 “Method0212“ entspricht der Methode “Nukleinsäure-Nachweis (z.B. PCR) aus Trachealsekret“). Die Angabe eines übergeordneten Feldes erfolgt durch die Referenzierung eines “IdField“ durch das Attribut “IdParent“. Die Bedeutung eines Feldes kann aus dem Attribut “GuiText“ erschlossen, und durch Informationen aus dem optionalen Feld “GuiToolTip“ präzisiert werden. Wird ein Feld innerhalb eines Kataloges genutzt, erfolgt mithilfe des Attributes “IdCatalogue“ ein Verweis auf diesen. Der Großteil der Felder ist jedoch keinem Katalog zugeordnet (entspricht einem “IdCatalogue“-Wert von 0).

   Es treten die folgenden strukturellen Besonderheiten auf:

3. Mehrfachdefinition[5] von Feldern

   Ein Feld wird genau so oft definiert, wie es einer Krankheit zugeordnet wird. Beispielsweise wird die Methode “Nukleinsäure-Nachweis (z.B. PCR) aus Trachealsekret“ mit dem FieldName “Method0212“ jeweils für die Krankheiten mit den Kürzeln “CVS“, “INV“ und “LEG“ angelegt.

   Weiterhin existieren:

[5] Als Mehrfachdefinition wird hier das mehrmalige Auftreten eines Elementes mit dem gleichen FieldName bezeichnet.

4. Krankheitsbezogene Spezialisierungen eines Feldes

   Die Felder können im Rahmen der Mehrfachdefinition unterschiedliche Werte des Attributes “IdParent“ annehmen. Dies führt dazu, dass ein Feld im Kontext einer Krankheit A eine andere Spezialisierung darstellt, als im Kontext einer Krankheit B. Ferner ist zu beachten, dass unter Umständen Elemente referenziert werden, die nicht Bestandteil der Terminologien sind. Das nachfolgende Listing 4.2 zeigt die Definitionen des Feldes “Method0212“ mit unterschiedlichen Werten des Attributes “IdParent“.

```
1 <Fields>
2   <Field IdField="115320" FieldName="Method0212" IdParent="
       1112" GuiText="Nukleinsaeure-Nachweis (z.B. PCR) aus
       Trachealsekret" IdCatalogue="0"/>
3   <Field IdField="137317" FieldName="Method0212" IdParent="
       137311" GuiText="Nukleinsaeure-Nachweis (z.B. PCR) aus
       Trachealsekret" IdCatalogue="0"/>
4   <Field IdField="138313" FieldName="Method0212" IdParent="
       138315" GuiText="Nukleinsaeure-Nachweis (z.B. PCR) aus
       Trachealsekret" IdCatalogue="0"/>
5 </Fields>
```

Listing 4.2: Mehrfachdefinitionen einer Methode mit veränderten IdParent-Werten

5. Eine Liste wird definiert als ein “Catalogue“ (Child-Element von “Catalogues“), wobei ein Listenelement einem “Catalogue“ untergeordnetem “Item“ entspricht. Die Identifizierung des jeweiligen Elementes wird erreicht durch die Attribute “IdCatalogue“ und “IdItem“, die Zuordnung eines Items zu einem Catalogue bildet das Attribut “IdCatalogue2Item“ ab. Ferner benennen “CatalogueName“ und “ItemName“ die entsprechenden Elemente. Der Inhalt eines Kataloges ist ebenfalls aus der Benennung (“CatalogueName“) abzuleiten (jeweils Konkatenation mit Krankheitskürzel):

   - “Form“ Krankheitsformen
   - “Pathogen“ Erreger
   - “Vaccine“ Impfstoffe
   - “Methode“ Methoden

   Das Listing 4.3 zeigt die weiteren Attribute “IsHierarchical“ (“1“ entspricht einem hierarchischen Katalog, “0“ entspricht einem Katalog ohne Hierarchien) und “IdParent“ (zur Referenzierung eines übergeordneten Items) am Beispiel des Kataloges “FormCLO“ (Krankheitsformen des Botulismus’).

```
<Catalogues>
  <Catalogue IdCatalogue="11001" CatalogueName="FormCLO"
     IsHierarchical="0">
    <Item IdCatalogue2Item="13343" IdItem="1002" ItemName="-
       nicht ermittelbar-" IdParent="0"/>
    <Item IdCatalogue2Item="13342" IdItem="1001" ItemName="-
       nicht erhoben-" IdParent="0"/>
    <Item IdCatalogue2Item="13344" IdItem="110010001"
       ItemName="Lebensmittelbedingter Botulismus" IdParent=
       "0"/>
    <Item IdCatalogue2Item="13345" IdItem="110010002"
       ItemName="SÃ€uglingsbotulismus" IdParent="0"/>
    <Item IdCatalogue2Item="13346" IdItem="110010003"
       ItemName="Wundbotulismus" IdParent="0"/>
  </Catalogue>
</Catalogues>
```

Listing 4.3: Katalogdefinition der Krankheitsformen des Botulismus'

6. Die Elemente "MigrationSurvNet3" (Unterelemente von "MigrationSurvNet3s") verbinden jeweils das Kürzel einer Krankheit mit den Schlüsseln ("IdField") der zugeordneten Felder.

```
<MigrationSurvNet3s>
  <MigrationSurvNet3 IdField="115320" FieldName="Method0212"
     Code="CVS"/>
  <MigrationSurvNet3 IdField="137317" FieldName="Method0212"
     Code="INV"/>
  <MigrationSurvNet3 IdField="138313" FieldName="Method0212"
     Code="LEG"/>
</MigrationSurvNet3s>
```

Listing 4.4: Zuordnung einer Methode zu den Krankheiten mit den Kürzeln CVS, INV und LEG

Die vollständige Datei mit dem Namen "SurvNetMeta.xml" befindet sich im Anhang auf der beigelegten CD.

4.5 Auswahl einer Untermenge

Die in Abschnitt 4.3 festgelegten Kompetenzfragen fordern eine Umsetzung nahezu aller beschriebenen Terminologie-Elemente. Mit Rücksicht auf die Ergebnisse der Export-Struktur-Analyse werden verschiedene Designentscheidungen getroffen, die im Folgenden erläutert werden.

Die Abbildung einer **Krankheit** erfolgt unter Ausschluss des Attributes "IfSGBundesland". Das sonstige verfügbare Wissen wird jedoch für die Abbildung herangezogen. Dabei wird die Ziffernschreibweise für die Angabe der Bundesländer, in denen eine Meldepflicht besteht, aufgelöst und die Menge der Bundesländer in einer Liste (ähnlich der Erreger- und Impfstofflisten) zusammengefasst. Durch dieses Vorgehen wird die Beantwortung von *CQ2* ermöglicht.
Die Hierarchien innerhalb der **Listen** werden vollständig abgebildet, auf eine Umsetzung der Hierarchien zwischen den Listen wird jedoch verzichtet, da dies nicht durch eine Anforderung spezifiziert wurde. Weiterhin werden die in jedem Katalog enthaltenen Elemente "nicht erhoben", "nicht ermittelbar" und "andere/sonstige" von der Abbildung ausgeschlossen.
Die vorhergehende Analyse offenbarte die Besonderheit der Mehrfachdefinitionen eines **Feldes**. Diese werden zugunsten der Bildung von Polyhierarchien aufgelöst.
Ein Codesystem ist häufig charakterisiert durch die Bildung von Klassen. So findet die Kategorisierung einer Prozedur im OPS-Codesystem beispielsweise durch die Einteilung in *Diagnostische Massnahmen, Bildgebende Diagnostik, Operationen*[6] und weitere Bereiche statt. Im Falle der hier untersuchten Terminologien lässt sich eine Einteilung vorrangig aus der Benennung der Felder ableiten. Lediglich eine Krankheit ist eindeutig dem XML-Element "Diseases" untergeordnet. In Anlehnung an die Beantwortung von *CQ1* wird somit die Kategorie "Krankheiten" eingeführt.

[6] `http://www.dimdi.de/static/de/klassi/ops/kodesuche/onlinefassungen/opshtml2013/index.htm`, letzter Zugriff am 25.08.13

4.6 Definition der Abbildung

Der folgende Abschnitt beschreibt die Abbildung der Terminologie-Elemente gemäß der CTS2-Spezifikation. Dabei ist zu beachten, dass hier keine Umsetzung einer Ontologie im eigentlichen Sinne vorgenommen wird. Es erfolgt beispielsweise keine Bildung der Klassen “Krankheit“, “Methode“ oder “Symptom“, welche formal innerhalb einer T-Box beschrieben werden. Dieses terminologische Wissen ist gegeben durch die RDF-Signaturen, welche beispielsweise eine *ClassDescription* als ein Konzept definieren. Die Elemente des XML-Inputs (Krankheiten, Symptome usw.) entsprechen konkreten Instanzen dieses Konzeptes.[7]

Die Abbildungsvorschrift wird unterteilt in die

1. Definition der Begriffe und
2. Definition der Relationen und Assoziationen.

Zur Veranschaulichung dieses Vorgangs wird Bezug genommen auf die in Abschnitt 2.2.2 genutzte Darstellung des Information Models. Eine konkrete Instanziierung eines Begriffes K kann dann mithilfe des folgenden Schemas beschrieben werden:

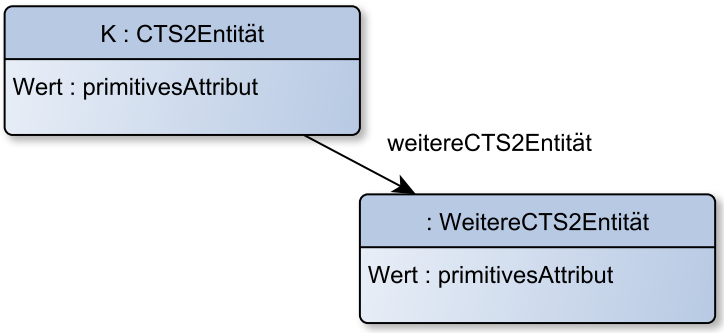

Abbildung 4.4: Schema zur Erläuterung der Definition

K kann als Instanz von “CTS2Entität“ mit dem Attribut “primitivesAttribut“ angenommen werden. Ein primitives Attribut ist dadurch gekennzeichnet, dass es keiner eigenen Entität laut CTS2-Spezifikation entspricht, sondern eher einem primitiven Datentyp (z.B. dem Datentyp String) gleicht. Eine Kante hingegen repräsentiert ein Attribut, welches einer eigenen CTS2-Klasse entspricht. Im Sinne der Repräsentation mittels RDF kann diese Kante ebenfalls als Relation zu einer Instanz eben jener Entität verstanden werden. K ist nun über die Relation mit dem Namen “weitereCTS2Entität“ verbunden mit einer anonymen Instanz (erkennbar am fehlenden Bezeichner der Entität) von “WeitereCTS2Entität“.

[7] Zur Abgrenzung werden in diesem Kapitel Krankheiten, Symptome, Methoden usw. als Begriffe und Entitäten des CTS2-Models als Klassen bezeichnet.

1.) Definition der Begriffe

Die Abbildung eines Begriffes - unabhängig von strukturellen oder inhaltlichen Eigenschaften - umfasst in jedem Fall die Entitäten *ClassDescription, ScopedEntityName* und *Designation*. Durch die Verwendung dieser 3 Bausteine wird sowohl eine exakte Identifizierung innerhalb des Systems, als auch eine Einbeziehung in die Suchinhalte garantiert. Zum Anreichern mit zusätzlichen Informationen stehen weiterhin *Definitions* (für normierte Sachverhalte o.ä.) und *Notes* (geeignet für einfache Hinweise) zur Verfügung.

So wird der zentrale Begriff der Krankheit gemäß der Spezifikation als *ClassDescription* umgesetzt. Diese beschreibt eine RDF-Resource (*about*), welche durch das Jena-Framework erstellt wird.
Der Name der Krankheit wird angesehen als das primäre Beschreibungsmittel und wird daher sowohl dem Pflichtelement *preferredTerm* einer ClassDescription, als auch dem *value*-Attribut einer *Designation* zugeordnet. Die Designation kann durch die Angabe einer *designationRole* typisiert werden. Diese wird als PREFERRED festgelegt[8].

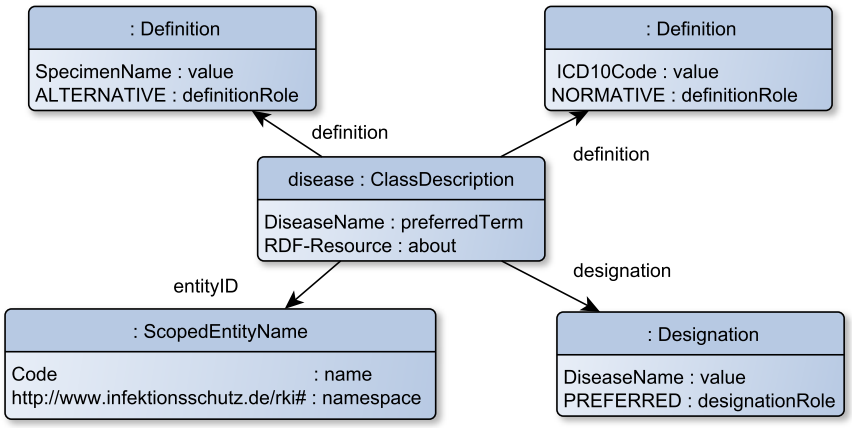

Abbildung 4.5: Abbildung des Konzepts einer Krankheit

Die Informationen ICD-Code und der Name des Erregers erfüllen den Charakter einer Definition, wobei ersteres durch die *definitionRole* als NORMATIVE, letzteres als ALTERNATIVE deklariert wird. Die eigentlichen Werte werden den *value*-Attributen der *Definition*-Entitäten zugewiesen. Die Abbildung 4.5 veranschaulicht die erläuterte Definition.

Um eine Resource eindeutig innerhalb einer Codesystemversion zu identifizieren, muss die Zuweisung einer *entityID* erfolgen. Durch die Verknüpfung eines Namensraumes (*namespace)*

[8] Weitere Möglichkeiten einer Typisierung sind ALTERNATIVE und HIDDEN. Der jeweilige Einsatz ist wird in der Tabelle A.2 im Anhang erläutert.

und eines Namens (*name*) in einem *ScopedEntityName* wird diese Anforderung erfüllt. Für die momentane Version des Codesystems wird der Namensraum `http://www.infektionsschutz.de/rki#` festgelegt. In Verbindung mit dem Kürzel einer Krankheit (*Code*) ergibt dies ein eindeutiges Konstrukt.

Die Abbildung der Felder erfordert ebenfalls die Instanziierung einer *ClassDescription*, wobei als *preferredTerm* der Gui-Text der Felder verwendet wird. Dieser entspricht gleichzeitig dem *value*-Attribut der mit PREFERRED typisierten *Designation*. Da über die Felder keine normierten Aussagen vorliegen, sondern eher weiterführende Hinweise gegeben wurden, fiel die Auswahl der Repräsentation dieses Wissens auf eine *Definition* mit der *designationRole* INFORMATIVE.

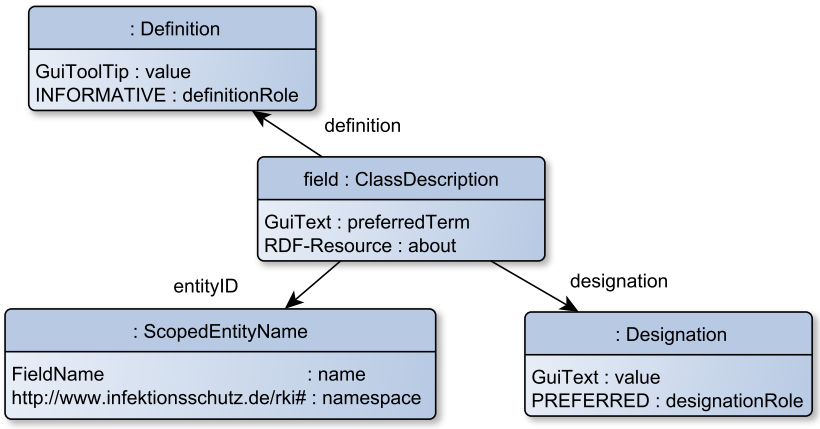

Abbildung 4.6: Abbildung der Felder

Die Erstellung einer entsprechenden RDF-Resource (*about*-Attribut) und die Festlegung des FieldNames als *name*-Attribut des *ScopedEntityNames* vervollständigen diese Definition.

Wie die vorhergehenden Analysen gezeigt haben, handelt es sich bei Listen und Listenelementen um vergleichsweise wenig beschriebene Konzepte. Es genügt daher, die in der XML-Struktur notierten Attribute auf die 3 wesentlichen Entitäten *ClassDescription, Designation* und *ScopedEntityName* zu projizieren. Die Ergebnisstruktur zeigt die Abbildung 4.7.

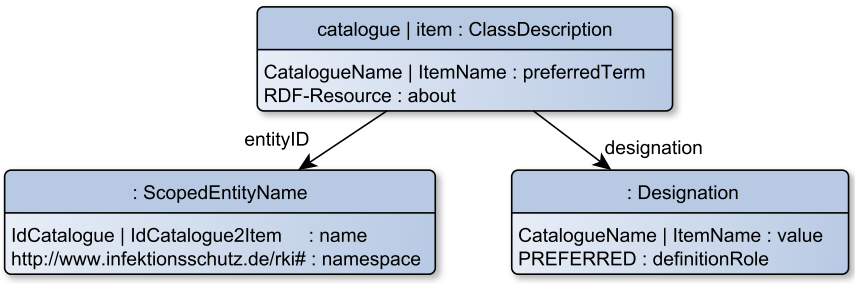

Abbildung 4.7: Definition der Abbildung von Listen und Listenelementen

Die Tabelle A.3 im Anhang fasst die Abbildung der Begriffe zusammen.

2.) Definition der Relationen und Assoziationen

Im Hinblick auf die Umsetzung der hierarchischen Strukturen und der Beziehungen zwischen den instanziierten Elementen werden die folgenden Prädikate notwendig:

- “subClassOf“ Abbildung von Hierarchien
- “hasPathogen“ Zuweisung von Erregern
- “hasVaccine“ Zuweisung von Impfstoffen
- “hasForm“ Zuweisung von Krankheitsformen
- “hasSymptom“ Zuweisung von Symptomen
- “hasMethod“ Zuweisung von Methoden
- “isNotifiableIn“ Angabe der Bundesländer, in denen eine Krankheit meldepflichtig ist

Diese entsprechen in einer Aussage der Form *Subjekt - Prädikat - Objekt* dem “Bindeglied“ zwischen Quell- und Zielentität.

Zur Umsetzung dieses Konstrukts stellt die CTS2-Spezifikation die Entität *PredicateDescription* zur Verfügung. Ebenso wie die ClassDescription ist die PredicateDescription eine Unterklasse von EntityDescriptionBase, d.h. die Entitäten sind hinsichtlich ihrer Attribute identisch (siehe Abbildung A.2 im Anhang). Es ergibt sich demnach folgende Abbildungsvorschrift:[9]:

[9] Die oben verwendete Bezeichnung des Prädikates entspricht dem “predicateName“. Im Falle des “subClassOf“-Prädikates erfolgt eine entsprechend veränderte Angabe des Namensraumes.

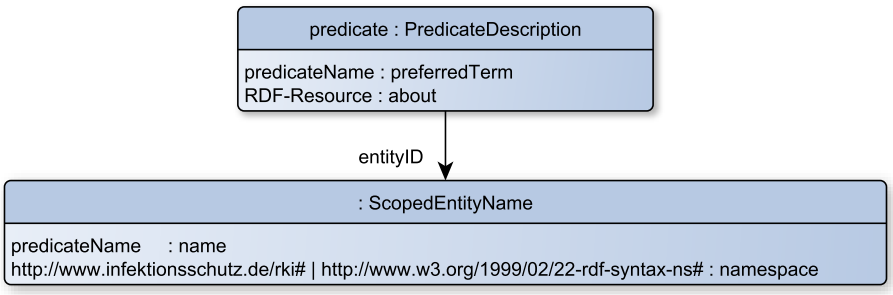

Abbildung 4.8: Definition der Relationen

Die eigentliche Vernetzung erfolgt mithilfe der Entitäten *Association* und *StatementTarget* sowie der damit verbundenen Relationen *subject, predicate, target* und *entity*. Gemäß der in Abschnitt 2.2.2 dargestellten Abbildungsvorschrift entsprechen diese Relationen *rdf:Properties*, deren zulässige Definitions- und Wertebereiche durch die in Anhang A.2 dargestellten RDF-Signaturen festgelegt sind.

Demnach können die Relationen *subject, predicate* und *entity* mit Instanzen der Klasse *EntityDescriptionBase* (bzw. deren Unterklassen) verbunden werden. Ebenfalls konsistent ist eine Assoziation, wenn deren Property *target* auf eine Instanz der Klasse *StatementTarget* verweist.

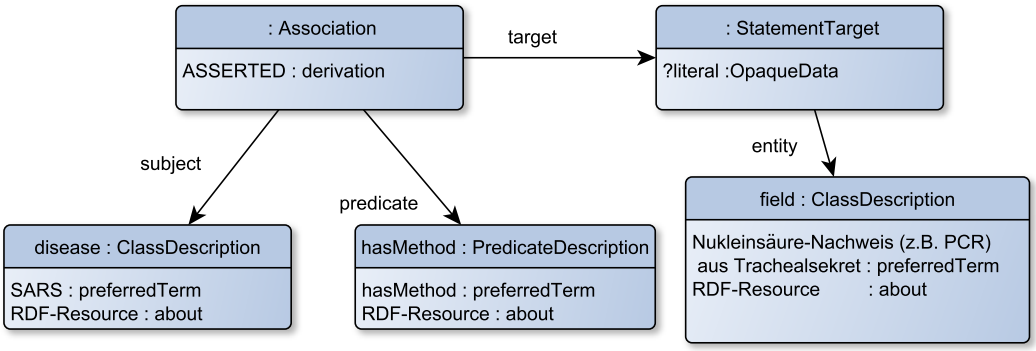

Abbildung 4.9: Definition einer *Association*

Die Abbildung 4.9 zeigt an einem Beispiel die Erstellung einer *Association*[10]: Die Krankheit *SARS* mit dem Kürzel CVS kann nachgewiesen werden durch einen "Nukleinsäure-Nachweis (z.B. PCR) aus Trachealsekret" (gekennzeichnet durch den FieldName "Method0212"). Man gehe davon aus, dass sowohl die Krankheit, als auch die Methode bereits als konkrete Instanzen einer *ClassDescription* im RDF-Store enthalten sind. Nun folgt die Erstellung einer Instanz der Klasse *Association*, wobei dem Attribut *derivation* der Wert ASSERTED

[10]Die Darstellung ist beschränkt auf die für das Verständnis wesentlichen Relationen und Entitäten.

zugewiesen wird. Durch diesen wird ausgedrückt, dass die Aussage explizit hinzugefügt wurde[11].

Die Association verknüpft nun die Instanzen durch die folgenden Properties:

- **subject** Bei dem Subjekt der Aussage handelt es sich um die Instanz, welche die Krankheit SARS beschreibt. In der Abbildung entspricht diese der *disease : ClassDescription* mit dem *preferredTerm* SARS.
- **predicate** Da es sich bei dem Objekt um eine Methode handelt, wird als *predicate* die *PredicateDescription* mit dem *preferredTerm* hasMethod zugeordnet.
- **target** Die Schema-Definition beschränkt den Wertebereich dieser Property auf Instanzen der Klasse *StatementTarget*. Die Instanz selbst kann entweder auf ein Literal verweisen (dann referenziert sie den Typ *OpaqueData*), oder sie wird über die Kante *entity* mit einer ClassDescription verbunden.
- **entity** In diesem Beispiel wird der *entity*-Relation die Instanz einer ClassDescription (hier benannt mit *field*) mit dem *preferredTerm* "Nukleinsäure-Nachweis (z.B. PCR) aus Trachealsekret" zugewiesen.

Nach dieser Vorgehensweise werden alle Assoziationen erstellt. Es werden dabei stets Instanzen der Entität *ClassDescription* durch eine *PredicateDescription* vernetzt.

Daher erscheint die Definition der insgesamt 6 Prädikate, welche die Zuweisung eines Feldes oder einer Liste zu einer Krankheit realisieren, zunächst nicht sinnig. So würde die Definition eines Prädikates "hasField" diese Zuordnungen vereinigen. Um jedoch eine komfortablere Beantwortung der Kompetenzfragen mithilfe von SPARQL zu erreichen, wurde diese Unterteilung vorgenommen.

[11] Ist die Aussage Ergebnis eines Inferenzmechanismus', so würde dem Attribut der Wert INFERRED zugewiesen werden

4.7 Entwurf und Implementierung

Im Folgenden wird beschrieben, wie die Definition der Abbildung umgesetzt wurde.
Der Import einer Terminolgie in den RDF-Store macht die Implementierung eines Terminologie-Loaders notwendig. Diese in Java geschriebene Klasse übernimmt die Extraktion von Daten aus verschiedenen Formaten sowie die eigentliche Abbildung nach CTS2. So wurde beispielsweise die Abbildung des ICD-Codesystems mittels eines Claml[12]-Imports realisiert.

Solche Loader sind in der Business-Schicht einzuordnen (siehe Abbildung 2.2 in Kapitel 2.2.2), in der sie sich verschiedener Komponenten der darunter liegenden domänenunabhängigen Schicht bedienen. Zu diesen gehört unter anderem die **Jena-API** (siehe Kapitel 2.1.3.2), welche zur Erstellung und Verarbeitung der RDF-Graphen herangezogen wird. Ferner definieren die vom Fraunhofer FOKUS entwickelten **Negros-Utilities** Methoden zur Validierung hinzugefügter Inhalte anhand der RDF-Signaturen. Dies geschieht unter Zugriff auf eine leichtgewichtige **SPARQL-Engine**. Ein weiteres Konstrukt stellen vordefinierte **SPARQL-Templates** dar, welche durch spezielle **Bindings** individuell nutzbar gemacht werden können.

Der Prozess der Implementierung kann untergliedert werden in die 3 Subprozesse:

1. vorbereitende Maßnahmen
2. Verarbeitung des XML-Inputs
3. Nachbereitung

zu 1.) vorbereitende Maßnahmen
Ähnlich eines Zugriffs auf eine Datenbank, muss auch hier zunächst eine Verbindung zum RDF-Quad-Store hergestellt werden. Anschließend erfolgt die Festlegung verschiedener statischer Attribute (unter anderem Festlegen des Namensraumes, des Namen und der Beschreibung der Terminologie). Es folgt das Laden der Quell-Datei sowie der RDF-Signaturen. Um den Aussagen, welche im Zuge der Abbildung getätigt werden, ein geeignetes Behältnis zu geben, wird ein *WorkModel* erstellt. Dabei handelt es sich um eine von **Negros** bereitgestellte Klasse, die das Jena-Interface *Model* adaptiert und um zusätzliche Konstrukte erweitert. Abschließend erfolgt die Erstellung des Codesystems unter Zugriff auf die definierten Attribute, als auch die Erstellung der Prädikate.

[12] Die *Classification Markup Language* stellt ein XML-Format dar, welches besondere Anwendung beim Austausch von medizinischen Codesystemen findet. [Ngo12]

zu 2.) Verarbeitung des XML-Inputs

Die Verarbeitung der Datei erfolgt ebenfalls mit einer Negros-Komponente. Der *XMLStream-Traverser* greift auf die von Java angebotene *Streaming API for XML (StAX)*[13] zurück. Mit der Auswahl dieses Werkzeuges wird eine performante Curser-Verarbeitung erreicht, welche - im Gegensatz zu *Push-APIs* (beispielsweise SAX oder DOM) - als ein von der Anwendung gesteuertes Streaming angesehen werden kann. Das bedeutet, dass der Parser auf Anweisungen wartet und nicht die Anwendung zur Verarbeitung der gerade gefundenen Blöcke zwingt. Diese serielle Vorgehensweise erlaubt jedoch lediglich das Vorwärtsgehen innerhalb einer XML-Datei. Somit erfolgt zunächst die Abbildung der Krankheiten, anschließend werden die Feld- und Katalog-Definitionen umgesetzt. Der letzte Schritt der Verarbeitung besteht in der Implementierung der Assoziationen.

zu 3.) Nachbereitung

Nachdem das Vokabular vollständig abgebildet wurde, erfolgt die Validierung mithilfe der **SPARQL-Engine**. Ein Beispiel für einen groben Verstoß gegen das Schema visualisiert die folgende Abbildung.

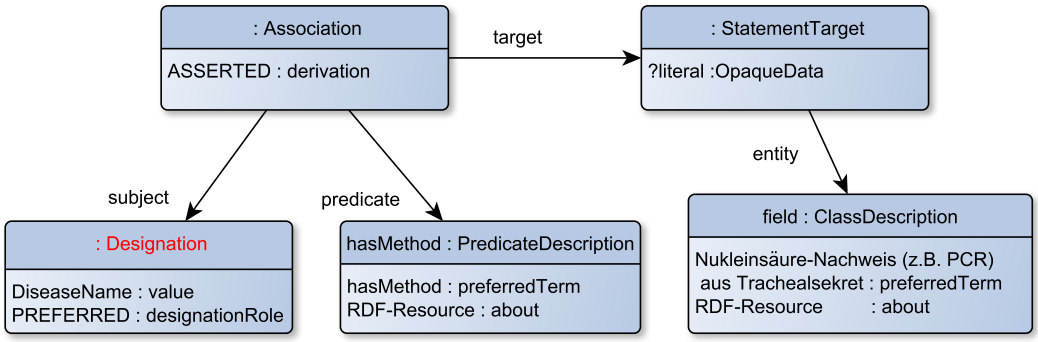

Abbildung 4.10: Beispiel einer nicht Schema-konformen Definition

Der Wertebereich der *subject*-Property ist auf Instanzen der Entität *EntityDescriptionBase* eingeschränkt. Somit wäre die Zuweisung der rot markierten *Designation* nicht Schema-konform.

Der nächste Schritt beinhaltet die Definition von *Toplevel-Klassen* (Klassen, welche keine Oberklasse haben), unter Zugriff auf das SPARQL-Template "constructTopLevelClasses"[14]. Toplevel-Klassen sind vorrangig relevant für die Darstellung eines Codesystems im Terminologie-Browser.

[13] `http://www.xml.com/pub/a/2003/09/17/stax.html`, letzter Zugriff am 30.08.13
[14] Die Datei "queriesRKI.ttl" beinhaltet die definierten SPARQL-Templates. Diese Datei befindet sich im Anhang auf der beigelegten CD.

Das validierte und nachbearbeitete Codesystem wird nun mit weiteren im Store befindlichen Terminologien vernetzt. Die Durchführung einer Volltextindizierung mithilfe von *Apache Lucene*[15] ermöglicht die Suche ähnlicher Konzepte. Zuletzt wird das WorkModel mitsamt der getätigten Aussagen dem RDF-Store hinzugefügt.

Der Quellcode der beschriebenen Implementierung befindet sich im Anhang auf der CD.

[15] `http://lucene.apache.org/core/`, letzter Zugriff am 30.08.13

5 Evaluierung der Abbildung

Im Rahmen der Anforderungsanalyse wurden verschiedene Kompetenzfragen erarbeitet, welche als Bewertungsmaßstab für die umgesetzte Abbildung verwendet werden.
Diese Fragen zielten zunächst darauf ab, die Grundfunktionalität (Abfrage von Wissen über hinzugefügte Instanzen und deren Relationen) zu überprüfen. Weiterhin sollte erfasst werden, in welchem Maße Verflechtungen mit anderen medizinischen Codesystemen ausgenutzt werden können. Bevor die Ergebnisse der Evaluierung vorgestellt werden, soll kurz das Vorgehen bei der Umsetzung der Kompetenzfragen erläutert werden.

SPARQL-Templates und Bindings

Die Abfrage der Terminologien erfolgt grundsätzlich mithilfe von SPARQL. Um die Beantwortung immer wiederkehrender, ähnlicher Fragestellungen zu erleichtern, steht eine Sammlung von Templates zur Verfügung. Durch die Zuweisung eines Wertes zu in einer Query verwendeten Variable entstehen individuelle Anfragen. Die folgende Beispiel-Query veranschaulicht dieses **Binding**:

```
[q:name "getClassDesignation";
 q:sparql
 """
 select ?preferredTerm ?code ?designation
  {
  ?clazz
    a :ClassDescription.
  ?clazz
    :preferredTerm ?preferredTerm.
  ?clazz
    :entityID ?h1.?h1 :name ?code.
  ?clazz
    :designation ?h2.?h2 :value ?designation.

  }
 """
].
```

Listing 5.1: SPARQL-Template “getClassDesignation“

Ohne ein vorheriges Binding resultiert die Ausführung dieser Query in einer Auflistung aller im RDF-Store vorhandenen *ClassDescriptions*. Dem *select*-Statement kann entnommen

werden, dass jeweils die Attribute *preferredTerm, code* sowie *designation* aufgeführt werden. Als unterstützendes Werkzeug kann bei der Formulierung einer Anfrage das Information Model herangezogen werden. So werden im Listing 5.1 genau die Tripel gefunden, welche auf die definierten Patterns passen. Demnach muss es eine Instanz einer *ClassDescription* geben, für welche ein *preferredTerm* festgelegt ist. Weiterhin muss diese *ClassDescription* über die Relationen *entityID* und *designation* verbunden sein mit Instanzen von *ScopedEntityName* (?h1) und *Designation* (?h2), deren Attribute *name* und *value* gebunden sind. Die Abbildung A.3 im Anhang veranschaulicht das Tripel-Pattern.

Mithilfe eines Bindings kann die Query nun angepasst werden. Beispielsweise würde eine Zuweisung eines Wertes zum *?preferredTerm* zu einer erheblichen Einschränkung der Ergebnistripel führen, da nun nur noch die *ClassDescriptions* mit dem entsprechenden *preferredTerm* ausgewählt werden.

Die im Rahmen dieser Arbeit verwendeten Templates enthält die Datei “queriesRKI.ttl“. Die Resultate können der Datei “competencyResults.ttl“ entnommen werden. Beide Dateien befinden sich auf der CD im Anhang.

Auswertung der Queries

Die Tabelle 5.1 zeigt die Auswertung der Kompetenzfragen. Für jede Frage sind der Name des jeweilig genutzten Templates sowie die getätigten Bindings festgehalten. Die letzte Spalte gibt an, ob die Frage erfolgreich beantwortet werden konnte.

Der Tabelle ist zu entnehmen, dass alle Fragestellungen erfolgreich von der Terminologie beantwortet werden konnten. Diese umfassten sowohl Anfragen an die primär umgesetzten Terminologien des Infektionsschutzes (CQ1 bis CQ8), als auch die Abfrage von Inhalten des ICD-10-Codesystems (CQ9 bis CQ11).

Tabelle 5.1: Auswertung der Kompetenzfragen

	Name der Query	getätigte Bindings	Beantwortung erfolgreich
CQ1	getChildClasses-r	preferredTermOfObject : "Diseases"	erfolgreich
CQ2	getRelations	preferredTermOfObject : “Nordrhein-Westfalen“ preferredTermOfPredicate : “isNotifiableIn“	erfolgreich
CQ3	getRelations	preferredTermOfSubject : “Campylobacter-Enteritis“ preferredTermOfPredicate : “hasSymptom“	erfolgreich
CQ4	getRelations	preferredTermOfSubject : “Ornithose“ preferredTermOfPredicate : “hasMethod“	erfolgreich
CQ5.1	askRelation	preferredTermOfSubject : “Diphtherie“ preferredTermOfPredicate : “hasVaccine“	erfolgreich
CQ5.2	getRelations	preferredTermOfSubject : “Diphtherie“ preferredTermOfPredicate : “hasVaccine“	erfolgreich
CQ6	getPathogensBy Symptom	preferredTermOfSymptom : “Dyspnoe (Atemstörung)“ symptomPredicate : “hasSymptom“ pathogenPredicate : “hasPathogen“	erfolgreich
CQ7	askRelation	preferredTermOfSubject : “EHEC-Erkrankung“ preferredTermOfPredicate : “hasSymptom“ preferredTermOfObject : “Fieber“	erfolgreich
CQ8	getRelations	preferredTermOfSubject : “Milzbrand“ preferredTermOfPredicate : “hasForm“	erfolgreich
CQ9	getICDByDisease	preferredTermOfDisease : “Botulismus“ resourceID : “ICD10de“	erfolgreich
CQ10	getICDSubClasses ByDisease	preferredTermOfDisease : “Fleckfieber“ resourceID : “ICD10de“	erfolgreich
CQ11	askRelation	icdCode : “B36“ preferredTermOfPredicate : “subClassOf“ preferredTermOfObject : “Diseases“	erfolgreich

6 Fazit

In den vergangenen Jahren wurde die Arbeit mit strukturierten Begriffssystemen zunehmend attraktiver. Dies ist nicht zuletzt zurückzuführen auf erfolgreiche Forschungsaktivitäten, die sich diesem Bereich widmen.

Die Arbeit beschreibt die Definition und Implementierung einer CTS2-standardisierten Abbildung von Terminologien der Domäne Infektionsschutz. Grundlage dafür ist das vom Fraunhofer Fokus entwickelte CTS2-Le-System, welches einen Ansatz zur Speicherung, Strukturierung und Verwaltung von Terminologien mithilfe semantischer Technologien präsentiert. Es erfolgte eine inhaltliche und strukturelle Untersuchung der verfügbaren Daten.
Die anschließend ausgewählte Teilmenge ließ die Konstruktion eines kompakten Begriffssystems zur Beschreibung von meldepflichtigen Krankheiten zu. Im Rahmen der Anforderungsanalyse erfolgte die Ausarbeitung sinnvoller Kompetenzfragen sowie die Formulierung dieser mithilfe von SPARQL-Templates. Diese dienten der Evaluierung der umgesetzten Abbildung und konnten vollständig und korrekt beantwortet werden. Ferner demonstrierten sie die unkomplizierte Ausnutzung weiterer Terminologie-Quellen am Beispiel des ICD-Codesystems.

6.1 Ausblick

Die Arbeit veranschaulichte bereits anhand einer eingeschränkten Menge, dass das System in der Lage ist, die Anforderungen, die an die moderne Terminologieverwaltung gestellt werden, zu erfüllen.
Daher liegt die Vermutung nahe, dass der Einsatz dieser Technologien in einem größeren Rahmen durchaus gewinnbringend ist. Zeigen wird sich dies im DEMIS-Projekt, welches eine Abbildung der vollständigen Terminologiemenge vorsieht. Der Import erfolgt ebenfalls mittels einer XML-Datei. Eine Alternative dazu wäre die gezielte Extraktion von Inhalten durch SQL-Anfragen an die Datenbank des Robert Koch-Instituts. Diese Vorgehensweise hätte den Vorteil, dass für die Abbildung nicht relevante Elemente von vornherein gefiltert werden würden.

Das im Rahmen der Arbeit verwendete System CTS2-Le wird kontinuierlich vom Institut weiterentwickelt. Zukünftige Zielsetzungen sind dabei u.a. die Implementierung von Service-Schnittstellen zur Anbindung von Clients und zur Erleichterung der Aktualisierungsvorgänge eines Codesystems.
Weitere interessante Untersuchungen könnten den Nutzen von Inferenzmechanismen oder ausdrucksstärkeren Beschreibungssprachen (z.B. der Web Ontology Language) zur Entwicklung von komplexen Ontologien beleuchten.

6.2 Schlusswort

Die vorliegende Arbeit hat gezeigt, dass eine Verknüpfung des CTS2-Standards mit semantischen Technologien eine zügige Entwicklung von Terminologien ermöglicht. Des Weiteren können ohne aufwändige Mapping-Verfahren gleiche Konzepte in benachbarten Codesystemen identifiziert und somit mächtige Wissensnetze gebildet werden.
Besonders der medizinische Sektor, in dem ein eindeutiges Vokabular und der Einsatz von Codesystemen zwingend erforderlich sind, profitiert von dieser Art der Wissensrepräsentation. Die nächsten Jahre werden zeigen, ob und in welchem Umfang solche Systeme zur Optimierung von medizinischen Prozessen beitragen werden.

Abkürzungsverzeichnis

CQ	Kompetenzfragen
CTS2	Common Terminology Services Release 2
DEMIS	Deutsches Elektronisches Meldesystem für Infektionsschutz
HL7	Health Level 7
ICD	Internationale statistische Klassifikation der Krankheiten und verwandter Gesundheitsprobleme
OMG	Object Management Group
OPS	Operationen- und Prozedurenschlüssel
PIM	Platform Independent Model
PSM	Platform Specific Model
RDF-Schema	Resource Description Framework Schema

Anhang

A Abbildungen und Tabellen

Abbildung A.1: Erreger-Liste der Adenovirus - K(eratok)onjunktivitis

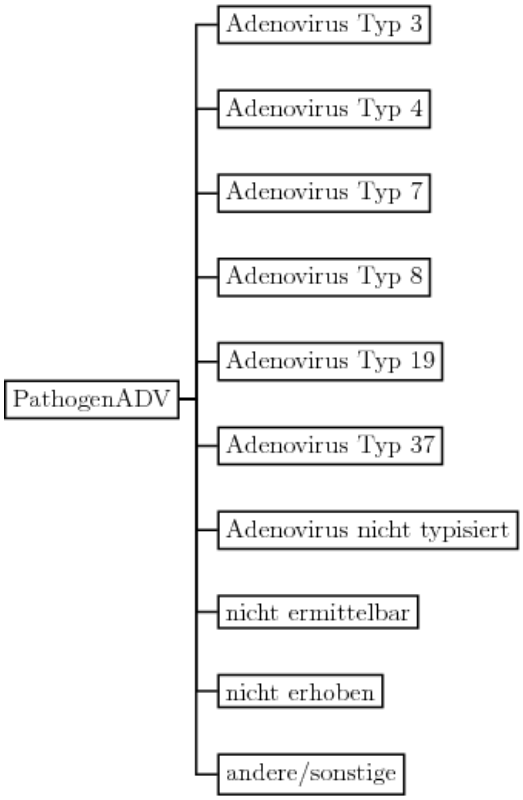

Tabelle A.1: CTS2 Dienst-Profile

Name	Erläuterung
Read	direkter Zugriff
Query	Suche und Exploration
Import/Export	Hinzufügen/Extrahieren eines anderen Dateiformates
Update	inkrementelles Aktualisieren
History	Verändern der Historie
Temporal	Zustandsabfrage
Maintenance	Einrichten eines inkrementellen Aktualisierungsvorganges

[Obj11a]

Tabelle A.2: Erläuterung der CTS2-Entitäten des Entity Description Models

Entität	Verwendung/Einsatz der Entität
EntityDescriptionBase	eine abstrakte Entität zum Treffen von Aussagen über Konzepte, Relationen und Individuen
ClassDescription	Definition eines Konzeptes
PredicateDescription	Definition einer Relation
ScopedEntityName	benennt eine Entität eindeutig innerhalb einer CodeSystem-Version
Designation	eine Zeichenkette in natürlicher Sprache für die Beschreibung eines Konzeptes oder einer Relation [1]; durch *Designation-Roles* Typisierung ermöglicht in PREFERRED (bevorzugte Beschreibung), ALTERNATIVE (alternative Beschreibung) oder HIDDEN (zur Präsentation nicht geeignete Beschreibung, jedoch für Suchanfragen verwendbar)
Definition	eine Zeichenkette in natürlicher Sprache zur Erklärung eines Konzeptes oder einer Relation; durch *Definition-Roles* Typisierung ermöglicht in INFORMATIVE (aufschlussreiche, wissenswerte Definition) oder NORMATIVE (gemäß einer Regel oder Norm definiert)
Note	ein Hinweis zu einem Konzept oder einer Relation
Association	eine Aussage über eine Relation zwischen einem Subjekt und einem Objekt; durch *derivation* Angabe, ob eine Aussage manuell (ASSERTED) oder durch Schlussfolgerungen (INFERRED) hinzugefügt wurde
StatementTarget	eine Aussage zum Zielobjekt einer Relation
OpaqueData	"undurchsichtiges Datum", repräsentiert Inhalte (Text oder eine formale Datenstruktur), die dem CTS2-System unklar sind
EntryDescription	eine Entität, die eine Resource beschreibt bzw. dabei hilft, diese zu identifizieren

[Obj11b]

[1] identisch mit einem Lexical Label im Simple Knowledge Organization System (SKOS)

```
@prefix : <http://isst.fraunhofer.de/sbi/cts2infoModel/signatures#>.
@prefix graphs: <http://isst.fraunhofer.de/sbi/cts2infoModel/graphs#>.
@prefix im: <http://isst.fraunhofer.de/sbi/cts2infoModel#>.


@prefix sig:  <urn:negros:signatures#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#> .


### ...


:Designation

  a rdfs:Class;
  rdfs:subClassOf :EntryDescription;

  sig:propertyConstraint [sig:onProperty  :designationRole;
     sig:range xsd:string;
     sig:min 1; sig:max 1;
     sig:facet
      '(PREFERRED)|(ALTERNATIVE)|(HIDDEN)';
  ].




:EntityDescriptionBase

  a rdfs:Class;
  sig:isAbstract true;
  sig:propertyConstraint [sig:onProperty  :about;
     sig:min 1;
     sig:max 1;
  ];
  sig:propertyConstraint [sig:onProperty  :entityID;
     sig:range :ScopedEntityName;
     sig:min 1; sig:max 1;
  ];
  sig:propertyConstraint [sig:onProperty  :alternateEntityID;
     sig:range :ScopedEntityName
  ];
  sig:propertyConstraint [sig:onProperty  :designation;
     sig:range :Designation
  ];
  ### helper prop; the preferred term from
  # designation's according to
  # CodeSystemVersionCatalogEntry
  #    .defaultLanguage
  # TODO: push up
  sig:propertyConstraint [sig:onProperty  :preferredTerm;
     sig:range xsd:string;
```

```
55      sig:min 1; sig:max 1;
56    ];
57    # for bypassing complete associations
58    sig:propertyConstraint [sig:onProperty  :childOf;
59      sig:range :EntityDescriptionBase;
60    ];
61    ###
62    sig:propertyConstraint [sig:onProperty  :definition;
63      sig:range :Definition
64    ];
65    sig:propertyConstraint [sig:onProperty  :note;
66      sig:range :Note
67    ].


70  :NamedEntityDescription

72    a rdfs:Class;
73    rdfs:subClassOf :EntityDescriptionBase.
```

Listing A.1: Signaturen der Entitäten *EntityDescriptionBase, Designation* und *NamedEntityDescription*

```
#----------------------------------

:StatementTarget

  ### XOR of the following properties

  a rdfs:Class;
  sig:propertyConstraint [sig:onProperty  :literal;
     sig:range :OpaqueData;
     sig:max 1;
  ];
  sig:propertyConstraint [sig:onProperty  :entity;
     sig:range :EntityDescriptionBase;
     sig:max 1;
  ].


:Association

  a rdfs:Class;
  sig:propertyConstraint [sig:onProperty  :subject;
     sig:range :EntityDescriptionBase;
     sig:min 1; sig:max 1;
  ];
  sig:propertyConstraint [sig:onProperty  :predicate;
     sig:range :EntityDescriptionBase;
     sig:min 1; sig:max 1;
  ];
  sig:propertyConstraint [sig:onProperty  :target;
     sig:range :StatementTarget;
     sig:min 1
  ];
  sig:propertyConstraint [sig:onProperty  :derivation;
     sig:range xsd:string;
     sig:min 1; sig:max 1;
     sig:facet
      '(ASSERTED)|(INFERRED)';
  ].

#----------------------------------
```

Listing A.2: Signaturen der Entitäten *StatementTarget* und *Association*

Tabelle A.3: Zusammenfassung der Abbildungsvorschrift für Begriffe

XML-Entität	Name des XML-Attributes	Name des CTS2-Attributes: CTS2-Entität
Disease	DiseaseName	preferredTerm : ClassDescription
	DiseaseName	value : Designation (mit designationRole = PREFERRED)
	Code	name : ScopedEntityName (*)
	SpecimenName	value : Definition (mit definitionRole = ALTERNATIVE)
	ICD10Code	value : Definition (mit definitionRole = NORMATIVE)
Field	GuiText	preferredTerm : ClassDescription
	GuiText	value : Designation (mit designationRole = PREFERRED)
	FieldName	name : ScopedEntityName (*)
	GuiToolTip	value : Definition (mit definitionRole = INFORMATIVE)
Catalogue	CatalogueName	preferredTerm : ClassDescription
	CatalogueName	value : Designation (mit designationRole = PREFERRED)
	IdCatalogue	name : ScopedEntityName (*)
Item	ItemName	preferredTerm : ClassDescription
	ItemName	value : Designation (mit designationRole = PREFERRED)
	IdCatalogue2Item	name : ScopedEntityName (*)

* mit namespace = http://www.infektionsschutz.de/rki#

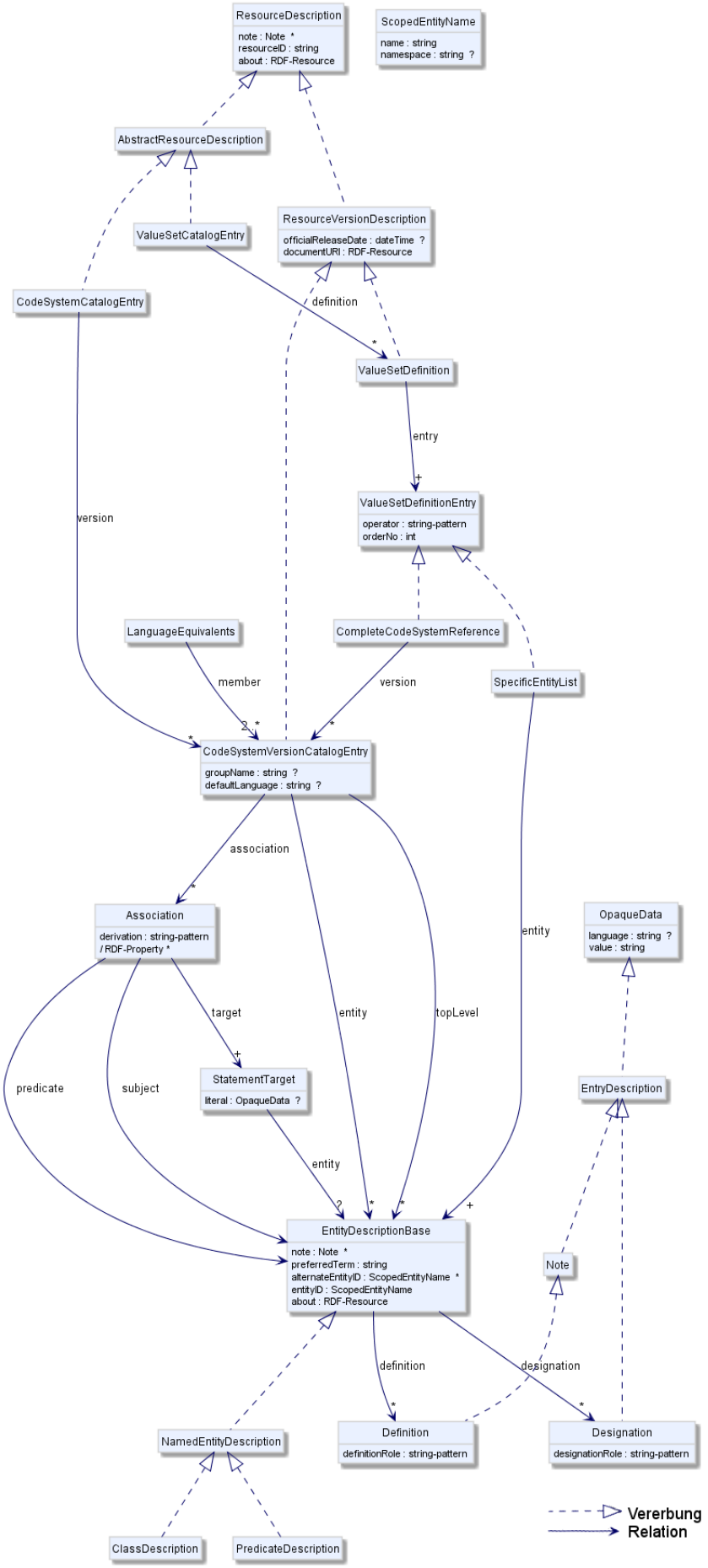

Abbildung A.2: Information Model des CTS2-Le-Systems

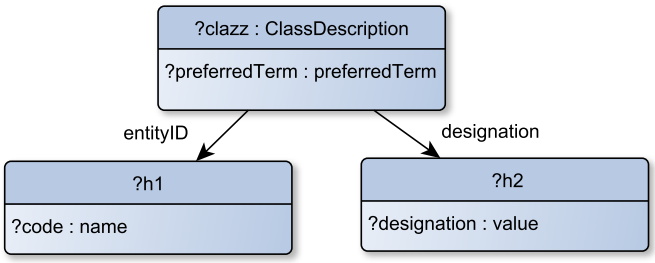

Abbildung A.3: Triple-Pattern der Query “getClassDesignation“

B Erläuterung der Datenbankentitäten des SurvNet@RKI-Systems

Zur Abbildung der Terminologien wird Bezug genommen auf die folgenden Entitäten:

1. Disease
2. Catalogue
3. Catalogue2Item
4. Item
5. Field

Disease

Die Datenbank-Entität *Disease* definiert eine Erkrankung durch zahlreiche Attribute, die in Tabelle B.1 inhaltlich und anhand der Krankheit Läuserückfallfieber auszugsweise beschrieben werden.

Tabelle B.1: Attribute der Entität Disease

Attributname	Inhalt	Werte des Läuserückfallfiebers
Code	Krankheitskürzel	BOR
DiseaseName	Name der Krankheit	Läuserückfallfieber
SpecimenName	Name des Erregers	Borrelia recurrentis
ICD10Code	Angabe des Codes im ICD10	A68.0

Catalogue, Catalogue2Item und Item

Diese Entitäten setzen die Abbildung von Katalogen und deren Elementen mithilfe folgender Attribute um.

Tabelle B.2: Attribute der Entitäten Catalogue, Catalogue2Item und Item

Attribute	Inhalt
Catalogue	Kataloge
IdCatalogue	Id des Kataloges
CatalogueName	Name des Kataloges (eindeutig)
IsHierarchical	Angabe, ob die Elemente des Kataloges hierarchisch angeordnet sind (1 entspricht einer Hierarchie, 0 entspricht einer flachen Anordnung)
Catalogue2Item	Zuordnung der Katalogwerte zu einem Katalog
IdCatalogue2Item	eindeutiger Schlüssel
IdCatalogue	Id des Kataloges
IdItem	Id des Katalogwertes
IdParent	Id des übergeordneten Katalogwertes (Verweis auf IdItem)
Item	Elemente der Kataloge
IdItem	Id des Katalogwertes
IdParent	Id des übergeordneten Katalogwertes (Verweis auf IdItem)
ItemName	Name des Katalogwertes

Field

Diese Datenbankentität dient generell der Abbildung von Feldern, die zwischen Programminstanzen transportiert werden.

Tabelle B.3: Attribute der Entität Field

Attribute	Inhalt
IdField	Id des Feldes
IdParent	Id des übergeordneten Feldes
IdCatalogue	Id des Kataloges (wenn Verweis auf diesen)
FieldName	Name des Feldes
GuiText	Beschreibung eines Feldes
GuiToolTip	ToolTip-Text für ein Feld

[Rob, S. 63–74]

Literaturverzeichnis

[BCM+10] Baader, F. ; Calvanese, D. ; McGuinness, D. ; Nardi, D. ; Patel-Schneider, P.: *The Description Logic Handbook*. Bd. Second Edition. Cambridge University Press, 2010

[Ben13] Benzler, Justus: DEMIS Deutsches Elektronisches Meldesystem für Infektionsschutz. In: *11.NRW-Dialog zum Infektionsschutz* Robert-Koch-Institut, 2013

[Bil13] Billig, A.: *Utilizing Semantic Technologies for a CTS2 Store*. `http://semantik.fokus.fraunhofer.de/WebCts2LE/main3/ini.jsp`. Version: 2013. – Fraunhofer FOKUS, CC eHealth; letzter Zugriff am 05.08.13

[DIMa] DIMDI Deutsches Institut für Medizinische Dokumentation und Information: *ICD-10-GM*. `http://www.dimdi.de/static/de/klassi/icd-10-gm/`. – letzter Zugriff am 25.08.13

[DIMb] DIMDI Deutsches Institut für Medizinische Dokumentation und Information: *Operationen- und Prozedurenschlüssel*. `http://www.dimdi.de/static/de/klassi/ops/index.htm`. – letzter Zugriff am 25.08.13

[Fen04] Fensel, D.: *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. Second Edition. Springer, 2004

[Gru93] Gruber, T. R.: A Translation Approach to Portable Ontology Specifications. In: *Knowledge Acquisition* 5(2) (1993), 199-220. `http://tomgruber.org/writing/ontolingua-kaj-1993.pdf`. – letzter Zugriff am 01.09.13

[Hah06] Hahn, Walther v.: *Terminus und Terminologielehre*. `http://nats-www.informatik.uni-hamburg.de/~vhahn/German/WortundBegriff/Terminologie.pdf`. Version: 2006. – letzter Zugriff am 15.08.13

[JFH03] J., Davies ; Fensel, D. ; Harmelen, F. van: *Towards The Semantic Web - Ontology-driven Knowledge Management*. John Wiley & Sons, LTD, 2003

[KLW95] Kifer, M. ; Lausen, G. ; Wu, J.: Logical Foundations of Object-Oriented and Frame-Based Languages. In: *Journal of the Association for Computing Machinery* (1995)

[MN01] McGuinness, D. L. ; Noy, N. F.: Ontology Development 101: A Guide to Creating Your First Ontology / Stanford University. 2001. – Forschungsbericht

[Nat] National Center for Biomedical Ontology: *CTS2 BioPortal wrapper*, `http://www.bioontology.org/wiki/index.php/CTS2_BioPortal_wrapper`. – letzter Zugriff am 14.08.13

[Nat12] National Cancer Institute (Hrsg.): *LexEVS Tool Overview*. National Cancer Institute, 2012. `https://wiki.nci.nih.gov/display/LexEVS/LexEVS`. – letzter Zugriff am 12.08.13

[Ngo12] Ngouongo, S.: *ClaML - Classification Markup Language - Einführung und Nutzung*. `http://www.imi.med.uni-erlangen.de/lehre/kolloquium/einzelne-nachricht/article/claml-classification-markup-language-einfuehrung-und-nutzung/`. Version: 2012. – letzter Zugriff am 30.08.13

[Obj11a] Object Management Group (Hrsg.): *Common Terminology Services 2*. FTF Beta 1. Object Management Group, September 2011

[Obj11b] Object Management Group (Hrsg.): *CTS2 Entity Description Services*. FTF Beta 1. Object Management Group, September 2011

[Rob] Robert-Koch-Institut (Hrsg.): *Fachkonzept für SurvNet@RKI 3.0*. Version 1.8. Robert-Koch-Institut, `http://survnet.rki.de`. – letzter Zugriff am 01.09.13

[Rob13] Robert-Koch-Institut (Hrsg.): *SurvNet@RKI – Das Meldesystem zum IfSG, Version 3.0.8*. Robert-Koch-Institut, 2013. `http://survnet.rki.de`. – letzter Zugriff am 30.08.13

[Sch05] Schöning, Uwe: *Logik für Informatiker*. Spektrum Akademischer Verlag, 2005

[SS08] Spreckelsen, C. ; Spitzer, K. ; Handels, H. (Hrsg.) ; Pöppl, S. (Hrsg.): *Wissensbasen und Expertensysteme in der Medizin*. Bd. 1. Auflage. Vieweg+Teubner, 2008

[Sta11] Stancl, C. ; Mayo Clinic (Hrsg.): *An Introduction to Common Terminology Services Release 2(CTS2)*. Mayo Clinic, November 2011

[The] The Apache Software Foundation: *Apache Jena*, `http://jena.apache.org/index.html`. – letzter Zugriff am 16.08.13

Abbildungsverzeichnis

Tabellenverzeichnis

Glossar

Frame Logic eine formale, logikbasierte Sprache zur Wissensrepräsentation, die sich an objektorientierten, frame-basierten Sprachen orientiert

Homonym ein Wort, welches für verschiedene Begriffe verwendet wird

Hypernym ein Oberbegriff

ICD dient als Klassifikationssystem zur Verschlüsselung von Diagnosen im ambulanten und stationären Bereich in Deutschland [DIMa]

OPS ein Klassifikationssystem zur Verschlüsselung von medizinischen Maßnahmen im stationären Bereich sowie beim ambulanten Operieren [DIMb]

Polyhierarchie eine Hierarchie, in welcher eine Klasse mehr als eine übergeordnete Klasse haben kann

Prädikatenlogik eine Erweiterung der Aussagenlogik um Quantoren, Funktions- und Prädikatsymbole [Sch05, S. 49]

RDF-Quad-Store eine Datenbank, die im Gegensatz zu einem Triple-Store zusätzliche Angaben zum Kontext eines Graphen speichert (bswp. ein sog. Named Graph)

Thesauri ein Wortnetz, in dem Begriffe durch Beziehungen miteinander verbunden sind

Value Set eine Untermenge von Werten aus einem Wertekatalog