

Bachelorarbeit

ERSTELLUNG EINER HL7 VERSION 3- SCHNITTSTELLE ZUR DATENKOMMUNIKATION IM INTERSEKTORALEN BEREICH AM BEISPIEL EINES VHITG-ARZTBRIEFES

Hochmuth, Steffen

geboren am 17. Oktober 1985 in Zwickau

Studiengang Informatik

Westsächsische Hochschule Zwickau

Fakultät Physikalische Technik / Informatik

Fachgruppe Informatik

Betreuer, Einrichtung: Prof. Dr. sc. hum. Anke Häber, WH Zwickau

Abgabetermin: 22.09.2010

Autorenreferat

Diese Bachelorarbeit wurde für das Erreichen des Informatikabschlusses Bachelor of Science an der Westfälischen Hochschule Zwickau erstellt.

Im ersten Teil dieser Arbeit wird ein Überblick über den derzeitigen Kommunikationsstandard im stationären Bereich, die HL7 Version 2, erarbeitet. Desweiteren wird der zurzeit nur spärlich im Einsatz befindliche Standard, die HL7 Version 3, zur Kommunikation zwischen dem stationären und dem ambulanten Bereich erläutert. Zusätzlich wird die Generierung von Schemadateien für HL7 Version 3 anhand eines Beispielszenarios erörtert.

Im zweiten Teil wird die Grundlage für die Erstellung der Transformationen von HL7 Version 2-Nachrichten in ein VHitG-Standardkonformes HL7 Version 3-Dokument geschaffen. Für eine HL7 Version 2-Nachricht wird in diesem Teil der Arbeit die Implementierung einer Transformation nach HL7 Version 3 in der Integrationsplattform Ensemble erläutert.

Danksagung

Hiermit möchte ich mich bei meiner Betreuerin Frau Professor Dr. sc. hum. Anke Häber, die mir immer mit sehr wertvollen Hinweisen und Ratschlägen kompetent und freundlich zur Seite stand, bedanken.

Desweiteren möchte ich mich bei Herr M. Sc. Thomas Nitzsche, Herr Dipl. Inf. (FH) Markus Lamprecht und Herr Dipl. Inf. (FH) Alexander Apel, die mir bei der Bewältigung der Aufgaben mit zahlreichen Informationen und Hilfestellungen beiseite standen, bedanken.

Auch möchte ich meinen großen Dank Herrn Dr. med. Walter Helm für die Korrektur der Arbeit aussprechen.

Inhaltsverzeichnis

1	Einleitung.....	1
1.1	Gegenstand und Problematik.....	1
1.2	Motivation	2
1.3	Problemstellung	2
1.4	Zielsetzung	2
2	Grundlagen.....	4
2.1	Die Intersektorale Versorgung	4
2.2	Das SCIPHOX-Projekt	6
2.3	Grundlegendes zum Health Level 7 (HL7)	6
2.4	Die Integrationsplattform Ensemble	7
3	Health Level 7 (HL7).....	8
3.1	HL7 Version 2	8
3.1.1	<i>Aufbau einer HL7 Version 2.6-Nachricht.....</i>	<i>8</i>
3.1.2	<i>HL7 Version 2.xml</i>	<i>9</i>
3.2	HL7 Version 3	10
3.2.1	<i>Grundlagen zur HL7 Version 3 Schemagenerierung</i>	<i>11</i>
3.2.2	<i>Generierung eines HL7 Version 3 Schemas anhand eines Beispielszenarios</i>	<i>19</i>
3.2.3	<i>Die Clinical Document Architecture (CDA).....</i>	<i>25</i>

4 Transformation einer HL7 Version 2-Nachricht in ein VHitG-Standardkonformes HL7 Version 3-Dokument.....	36
4.1 Anlegen der Produktion im Ensemble Management Portal	36
4.2 Erstellung der Transformationsklasse im Studio	38
4.2.1 <i>Prinzip der Transformation</i>	38
4.2.2 <i>Data Transformation Language (DTL)</i>	38
4.2.3 <i>Aufbau der Klasse zur Transformation</i>	41
4.2.4 <i>Workflow der Datentransformation</i>	42
5 Diskussion und Ausblick	43
5.1 Zielerreichung.....	43
5.2 Diskussion	43
5.3 Ausblick	44
6 Thesen	45

1 Einleitung

1.1 Gegenstand und Problematik

In der modernen intersektoralen Gesundheitsversorgung [iFunk06] findet sich eine komplexe netzartige Struktur von ambulanten und stationären Leistungserbringern. Dazu gehören unter anderen niedergelassene Ärzte, Labore, Apotheken, Kliniken und Kostenträger. Für die bestmögliche Patientenversorgung ist ein schneller und sicherer Informationsaustausch zwischen allen beteiligten Leistungserbringern von wesentlicher Bedeutung.

Im stationären Bereich hat sich Health Level 7 (HL7) [iHea10] als Standardsammlung für den Datenaustausch durchgesetzt. Die neueste Version von HL7 ist die 2005 zum ersten Mal publizierte eXtensible Markup Language (XML)¹-basierte HL7 Version 3, die sich von der HL7 Version 2 nicht nur im Format, sondern auch in Aufbau und Struktur unterscheidet. Die mit HL7 Version 3 einhergehende deutlich komplexere Struktur hat zur Folge, dass im stationären Bereich noch immer HL7 Version 2 als Standard gilt.

Im ambulanten Bereich finden sich viele verschiedene Computersysteme mit ihrer Vielzahl an eigenen Datenformaten.

Mithilfe der Integrationsplattform Ensemble von Intersystems sollen die Daten in dieser Arbeit aus den stationären in den ambulanten Bereich überführt werden. Ensemble kann zwar XML-Dateien und daher auch HL7 Version 3-Dateien einlesen, jedoch zur internen Verarbeitung ist momentan noch kein Schema für HL7 Version 3 in Ensemble vorhanden. Damit ist auch die Transformation von Daten in andere Formate nicht gewährleistet.

¹ XML ist eine strukturierte Beschreibungssprache, um Daten hierarchisch strukturiert in Textform darzustellen sie dient dem plattformunabhängigen Datenaustausch.

1.2 Motivation

Bezugnehmend auf die bisher genannten Angaben entstand der Gedanke, eine Routine zu entwickeln, mit deren Hilfe die Daten aus dem stationären in den ambulanten Bereich und auch zurück übertragen werden können, so dass sie jeweils im richtigen Format vorliegen. Um dies zu gewährleisten, soll mit dieser Arbeit eine Transformationskomponente in einer Integrationsplattform entwickelt werden, die Daten in das benötigte Format überführt. Da die Transformation zwischen dem standardisierten HL7 Version 3-Format und dem standardisierten HL7 Version 2-Format stattfindet, bietet sie den Vorteil, dass Komponenten, die diese Standards unterstützen, ausgetauscht werden können. Weitere Vorteile sind, dass die Transformation unabhängig von Benutzerinteraktionen arbeiten kann und die Daten deutlich schneller zur Verfügung stehen.

1.3 Problemstellung

Problem 1: HL7 Version 3 und HL7 Version 2 unterscheiden sich in Aufbau, Inhalt und Methodologie.

Problem 2: Ensemble von Intersystems kann zwar mit XML umgehen, jedoch wird HL7 Version 3 nicht direkt unterstützt.

Problem 3: Eine Transformation zwischen HL7 Version 3 und HL7 Version 2 ist nicht ohne weiteres möglich.

1.4 Zielsetzung

Ziel zu Problem 1:

Ziel 1.1: Unterschiede zwischen HL7 Version 3 und HL7 Version 2 im Bezug auf Aufbau, Struktur, Inhalt und Bedeutung.

Ziel zu Problem 2:

Ziel 2.1: Interne XML-Schemadefinition zum Verarbeiten von HL7 Version 3 Dokumenten in Ensemble.

Ziel zu Problem 3:

Ziel 3.1: Workflow zur Transformation und Durchleitung der Daten in Ensemble.

Ziel 3.2: Eine prototypische Routine zur Transformation und Durchleitung der Daten in Ensemble.

2 Grundlagen

2.1 Die Intersektorale Versorgung

Die ersten Ansätze zur Intersektoralen Versorgung gab es schon 1975, jedoch wurde erst mit der Gesundheitsreform 2000 der Ausbau der Intersektoralen Versorgung durch den Gesetzgeber begonnen^{[Pet06] Seite 194} und in den folgenden Jahren durch verschiedene Maßnahmen weiter vorangetrieben. Das Gesetz zur Modernisierung der gesetzlichen Krankenversicherung 2004 bildete die Grundlage der Verschmelzung der Grenzen zwischen den Krankenkassen und den Leistungserbringern. So können Einzelverträge auch ohne Billigung der Kassenärztlichen Vereinigungen zwischen Krankenkassen und Leistungserbringern geschlossen werden.

Unter der Intersektoralen Versorgung versteht man eine interdisziplinär-fachübergreifende Versorgung der Versicherten^{[Johner09] Seite 127}. Das bedeutet, dass die verschiedenen medizinischen Leistungssektoren miteinander vernetzt sind. Dies führt dazu, dass stationäre mit ambulanten Einrichtungen zusammenarbeiten und notwendige medizinische Leistungen für die vom Kooperationsvertrag betroffenen Patienten gemeinschaftlich erzielen (Abbildung I). Das hat zur Folge, dass eine Qualitätssteigerung der Patientenversorgung und eine Kostenreduzierung im Gesundheitswesen erzielt werden können.

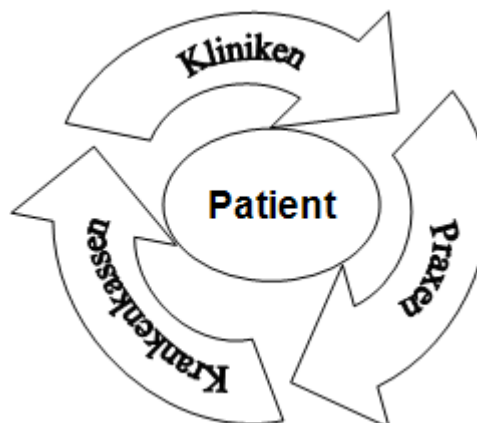


Abbildung I Prinzip der Intersektoralen Versorgung, eigene Darstellung

Dabei erhalten alle beteiligten Vertragspartner die Vergütungen für die erbrachten Leistungen von den Krankenkassen und teilen diese, abhängig vom Kooperationsvertrag, auf. Eine andere Möglichkeit, die sektoralen Grenzen zu überwinden, bieten die Medizinischen Versorgungszentren, in denen Ärzte aus unterschiedlichen Fachgebieten tätig sind ^[iKraka10].

Die Intersektorale Versorgung bietet mehrere Vorteile gegenüber der Regelversorgung. Dies sind unter anderem eine bessere Kommunikation und Koordination zwischen den Leistungserbringern. Das hat zur Folge, dass die Versorgung besser an die Bedürfnisse der Patienten angepasst ist. Da die erhobenen Daten untereinander einfacher ausgetauscht und damit unnötige Mehrfachuntersuchungen und -behandlungen vermieden werden können, stellt dies für den Patienten unter anderem eine geringere Belastung dar. Für die Leistungserbringer verringern sich zusätzlich die Gesamtbehandlungskosten und -zeiten gegenüber der Regelversorgung erheblich. Das führte in den letzten Jahren zu einer kontinuierlichen Erhöhung der Anzahl von abgeschlossenen Verträgen in der Intersektoralen Versorgung ^{[Pet06] Seite 195}.

Damit die Vorteile der Intersektoralen Versorgung zum Tragen kommen ist eine gute IT-Infrastruktur notwendig. Dies bedeutet, dass der Zugriff auf die erhobenen Daten für alle Vertragspartner zu jeder Zeit zu ermöglichen ist. Grundvoraussetzung hierfür ist, dass die betroffenen Einrichtungen miteinander verknüpft sind, und es müssen die Daten von allen betroffenen Parteien gelesen, ausgewertet und bearbeitet werden können. Da HL7 und die Formate im ambulanten Bereich nicht ohne weiteres ineinander überführt werden können, wurde das Projekt Standardized Communication of Information Systems in Physician Offices and Hospitals using XML (SCIPHOX) ^[iSciphox01] initiiert.

2.2 Das SCIPHOX-Projekt

Jahrelang wurden HL7 im stationären und xDT²^[iKBV06] im ambulanten Bereich parallel voneinander entwickelt. Da mit der Verbreitung der Intersektoralen Versorgung notwendig wurde, beide Bereiche näher zusammen zu führen, wurde 2000 das SCIPHOX-Projekt von der Arbeitsgemeinschaft SCIPHOX GbR mbH gestartet. Das Projekt sollte Vorgaben für Kommunikationsstandards, resultierend aus den gesammelten Informationen durch die Verwendung von HL7 und xDT, liefern und eine Symbiose zwischen den international geprägten HL7- und den national geprägten xDT-Standards darstellen. Das SCIPHOX-Projekt wurde durch Projekte mit ähnlichen Zielen unterstützt. Aufbauend auf den gewonnenen Erkenntnissen des SCIPHOX-Projektes wurde auch der VHitG-Arztbrief entwickelt ^[iBer07].

2.3 Grundlegendes zum Health Level 7 (HL7)

HL7 ist eine Gruppe internationaler Standards für den Datenaustausch im Gesundheitswesen, es stellt eine Verknüpfung zwischen den beteiligten Organisationen und deren IT-Systemen dar. Dabei steht die Zahl 7 für die 7. Schicht, die Applikationsschicht, des ISO/OSI-Referenzmodells^{[Pet06] Seite 132}. Dies bedeutet, dass HL7 sich mit der Kommunikation auf Anwendungsebene beschäftigt. Aktuell ist HL7 mit der Version 2 der Kommunikationsstandard im stationären Bereich. Mit der XML-basierten Version 3 soll dies auch auf den ambulanten Bereich übergreifen und somit eine bessere Kommunikation zwischen den beiden Bereichen ermöglichen ^{[iCorHe10] Seite 7}.

² xDT steht für die Schnittstellen im ambulanten Bereich dabei steht das x für einen Platzhalter. Zuerst waren diese Schnittstellen die Abrechnungsdatenträger (ADT). Daraus hat sich unter anderem in der folgenden Zeit der Behandlungsdatenträger (BDT) entwickelt.

2.4 Die Integrationsplattform Ensemble

Ensemble von der Intersystems GmbH ist eine Integrationsplattform^[Ensem10], das heißt, sie stellt eine Verbindung zwischen einem Daten-, einem Applikations- und einem Integrationsserver sowie einem Management-, dem Ensemble Management Portal, und einer Entwicklungsumgebung, dem Studio, her. Im Studio wird, wie in der Abbildung II zu sehen, die Implementierung der Caché ObjectScript-Klassen durchgeführt.

```

1 Class Bachelor.Operation.FileOut Extends Ens.BusinessOperation [ ClassType = "", ProcedureBlock
2 {
3
4 Parameter ADAPTER = "EnsLib.File.OutboundAdapter";
5
6 Parameter INVOCATION = "Queue";
7
8 Parameter SETTINGS = "Filename";
9
10 Property Filename As %String;
11
12 Method WriteToFile(pRequest As Bachelor.Message, Output pResponse As Ens.Response) As %Status
13 {
14 Do pRequest.Content.Rewind()
15 Set tFilename=.Adapter.CreateTimestamp(##Class(%File).GetFilename($Piece(pRequest.Source,
16
17 Set $ZTrap = "WriteToFileBT"
18 Set tStatus = ..Adapter.PutStream(tFilename, pRequest.Content)
19
20 Quit $$$OK
21
22 WriteToFileBT
23 Set $ZTrap = ""
24
25 Quit $$$ERROR($$$GeneralError, "An error has occurred in WriteToFile(" _ tFilename _ "): "
26 }
    
```

Abbildung II Das Ensemble Studio, eigene Darstellung

Das Ensemble Management Portal wird dabei über den Browser aufgerufen und hier lassen sich unter anderem die Produktionen anlegen. In der Abbildung III wird die Startseite des Ensemble Management Portals dargestellt.

Willkommen beim Ensemble Management Portal:
 Dies ist der Namespace STEHO.

Ensemble Troubled		Dienst	
Wiederherstellen		Uhrzeit der letzten Aktivität:	
Produktion: Bachelor.HL7V2ToV3 mehr...		Abgeschlossene Aktivitäten:	0
Startzeit:			
Job		Prozess	
Gesamtzahl der laufenden Jobs:	0	Uhrzeit der letzten Aktivität:	
Jobs OK:	0	Abgeschlossene Aktivitäten:	0
Inaktive Jobs:	0	Nachrichten in Bearbeitung:	0
Jobs mit anderem Status:	0	Nachrichten in Warteschlangen:	0
Protokoll		Operation	
Uhrzeit des letzten Fehlers:		Uhrzeit der letzten Aktivität:	
Gesamtzahl der Fehler:	0	Abgeschlossene Aktivitäten:	0
		Nachrichten in Bearbeitung:	0
		Nachrichten in Warteschlangen:	0

Aktualisierungsdauer in Sekunden: 21

Abbildung III Startseite des Ensemble Management Portals, eigene Darstellung

3 Health Level 7 (HL7)

3.1 HL7 Version 2

Die HL7 Version 2 ist zurzeit der Kommunikationsstandard im stationären Bereich. Der derzeit aktuellste veröffentlichte HL7 Version 2 Standard ist die Version 2.6. Der Nachfolger, die Version 2.7, ist derzeit noch in der Entwicklungsphase. Die Version 2.3.1 ist neben der Version 2.3 die meistgenutzte HL7 Version^{[iCorHe10] Seite 2} im stationären Bereich. Da die HL7 Version 2.6 derzeit die neueste HL7 Version ist, werden die Ausführungen und Beispiele im weiteren Verlauf dieser Arbeit sich daher auf die HL7 Version 2.6 beziehen.

3.1.1 Aufbau einer HL7 Version 2.6-Nachricht

Eine HL7 Version 2-Nachricht gliedert sich in Segmente, Felder, Komponenten und Subkomponenten. Dabei wird jedes Segment durch einen Zeilenumbruch von dem darauffolgenden Segment getrennt. Jedes Segment besitzt eine für jedes Segment genau definierte Anzahl an Feldern. Komponenten beziehungsweise Subkomponenten stellen dabei den Inhalt der Felder dar. Dabei haben die verschiedenen Felder einer HL7 Version 2-Nachricht unterschiedliche Trennzeichen: So werden standardmäßig Felder durch „|“, Komponenten durch „^“ und Subkomponenten durch „&“ voneinander getrennt. Die Struktur der HL7 Version 2-Nachrichten verdeutlicht die Abbildung IV am Beispiel eines MSH-Segments [iHL7Stan10].

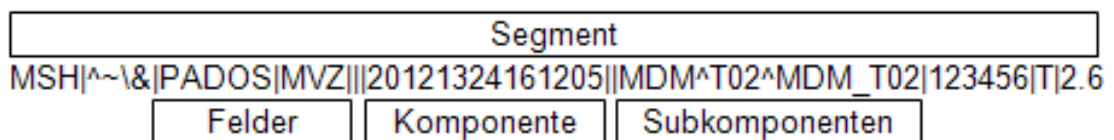


Abbildung IV *Aufbau eines HL7 Version 2-MSH-Segments, eigene Darstellung*

Jede HL7 Version 2-Nachricht besteht aus einem Headersegment. Dies ist immer das erste Segment einer HL7 Version 2-Nachricht. Im Nachrichtenheader, dem sogenannten „MSH“-Segment, stehen verschiedenste administrative Daten zu der Nachricht, unter anderem auch der Nachrichtentyp. Der Nachrichtentyp definiert

zusammen mit dem Trigger Event, ob und gegebenenfalls an welcher Stelle die Segmente in der Nachricht zu stehen haben. In der Abbildung IV ist die Nachricht vom Typ Medical Document Management (MDM), das Trigger Event ist vom Typ T02 (Original document notification and content). Dabei gibt das erste Feld jeden Segments immer dessen Namen an.

3.1.2 HL7 Version 2.xml

HL7 Version 2.xml-Nachrichten sind HL7 Version 2-Nachrichten, die aus einer XML-Struktur bestehen, sie bilden daher eine Sonderform der HL7 Version 2-Nachrichten. Sie können als Zwischenschritt bei der Transformation von HL7 Version 2 zu HL7 Version 3 eingesetzt werden und somit den Transformationsprozess erleichtern. Das zu Grunde liegende Prinzip ist, dass man für alle verwendeten Komponenten, Felder und Segmente einzelne Elemente erstellt. Die Namen der Elemente in der HL7 Version 2.xml-Nachricht richten sich dabei nach der Platzierung und dem Namen des Felds aus der HL7 Version 2-Nachricht. In dem Codeausschnitt I wird die Struktur einer HL7 Version 2.xml-Nachricht für das in der Abbildung IV dargestellte MSH-Segment gezeigt ^[iHeal10].

```
1| <MSH>
2| <MSH.1>|</MSH.1>
3| <MSH.2>^~\&</MSH.2>
4| <MSH.3>
5| <HD.1>PADOS</HD.1>
6| </MSH.3>
7| <MSH.4>
8| <HD.1>MVZ</HD.1>
9| </MSH.4>
10| <MSH.7>
11| <TS.1>20121324161205</TS.1>
12| </MSH.7>
13| <MSH.9>
14| <CM_MSG.1>MDM</CM_MSG.1>
15| <CM_MSG.2>T02</CM_MSG.2>
16| <CM_MSG.3>MDM_T02</CM_MSG.3>
17| </MSH.9>
18| <MSH.10>123456</MSH.10>
19| <MSH.11>
20| <PT.1>T</PT.1>
21| </MSH.11>
22| <MSH.12>
23| <VID.1>2.6</VID.1>
24| </MSH.12>
25| </MSH>
```

Codeausschnitt I *Aufbau des MSH-Elements einer HL7 Version 2.xml-Nachricht*

Aus dem Codeausschnitt I wird ersichtlich, dass die Dokumentenstruktur der HL7 Version 2.xml-Nachricht eine bessere Lesbarkeit und ein besseres Abfragen der Daten gewährleistet, da unbenutzte Felder entfernt wurden. So müssen nicht, wie in einer HL7 Version 2-Nachricht, die Elemente zum Auslesen nacheinander abgefragt werden, sondern die Elemente können direkt vom Wurzelement aus einzeln abgefragt werden.

3.2 HL7 Version 3

Ein Patient durchläuft in der Patientenversorgung viele unterschiedliche Stationen und hat Kontakt mit vielen beteiligten Personen, dabei entstehen Dokumente von der Anamnese, bei der Aufnahme, über OP-Berichte bis hin zu Überweisungen. In der Regel liegen diese Dokumente in unterschiedlichen Formaten, wie zum Beispiel Word, PDF und Excel, vor ^[iTon10] Seite 6. Diesem Formatbruch soll die internationale Clinical Document Architecture (CDA) entgegen wirken, so dass eine bessere Lesbarkeit und damit einhergehend eine bessere Auswertbarkeit gewährleistet werden kann. Der VHitG³ hat den Implementierungsleitfaden für den, auf der CDA Release 2 basierenden, elektronischen Arztbrief für die Bundesrepublik Deutschland herausgebracht ^[iVHitGbrief10].

HL7 Version 3 kann sowohl im stationären wie auch im ambulanten Bereich eingesetzt werden und unterstützt transsektorale Prozessketten. Die CDA ist Teil der HL7 Version 3. Da sie wie die HL7 Version 3 auf XML basiert, kann jedes Dokument gegen die HL7 Spezifikationen validiert, außerdem können Standardvokabularen sowie -datentypen verwendet werden ^[iHCl10]. Ein weiterer Vorteil von XML ist, dass die Dokumente im Gegensatz zu zum Beispiel Datenbanken, Word-, Excel- oder PDF-Dokumenten auf fast jedem System (zum Beispiel PDA und Handy) gelesen und

³ VHitG ist der Verband der Hersteller von IT-Lösungen im Gesundheitswesen e.V. Er ist eine Interessenvertretung gegenüber Politik und Presse und dient zur Förderung des elektronischen Datenaustauschs und der –kommunikation in der intersektoralen Versorgung ^[iVHitG10].

bearbeitet werden können. Zusätzlich kann der Inhalt der XML-Datei durch Einbinden eines Stylesheets im Browser zum Beispiel formatiert dargestellt werden.

3.2.1 Grundlagen zur HL7 Version 3 Schemagenerierung

Der Grundstein für die Entwicklung des objektorientierten Kommunikationsstandards HL7 Version 3 bildet das Zusammenfügen der statischen Modelle (Informationsmodelle) mit dem dynamischen Modell (Interaktionsmodell) ^[iHCI10].

Das Reference Information Model (RIM)

Das RIM bildet die Basis für die weiteren, davon abgeleiteten Informationsmodelle. Im RIM werden alle relevanten Klassen, deren Attribute und Datentypen des Gesundheitswesens dargestellt. Das RIM ist ein überwiegend generisches Modell. Die vier RIM-Basisklassen sind die Handlungen, welche Tätigkeiten im Gesundheitswesen darstellen, an denen Einheiten in bestimmten Rollen beteiligt sind. Beziehungen zwischen den Handlungen oder Rollen verbinden diese jeweils miteinander. In der Abbildung V sind die Klassen mit Beispielen schematisch dargestellt.

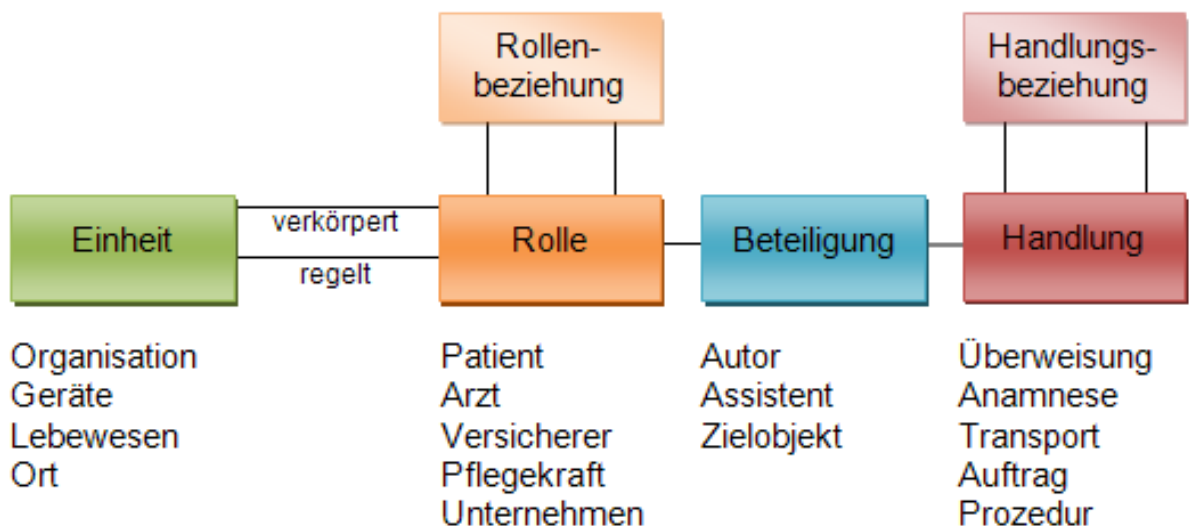


Abbildung V Die 4+2 Basisklassen des RIM, eigene Darstellung

Die Klassen sind folgendermaßen definiert ^[iHBA10]:

➤ Einheit

Die Einheit ist eine Klasse, die physikalische, für eine Handlung notwendige, Objekte beschreibt.

➤ **Rolle**

Die Rolle ist eine Klasse, die unabhängig von den Handlungen die Fähigkeiten einer Einheit beschreibt, dabei kann eine Einheit durch mehrere Rollen definiert werden.

➤ **Rollenbeziehung**

Rollenbeziehung ist eine Klasse, die die Abhängigkeiten zwischen Rollen beschreibt.

➤ **Beteiligung**

Die Beteiligung ist eine Klasse, die beschreibt, wie eine Einheit in einer bestimmten Rolle an einer Handlung teilnimmt.

➤ **Handlung**

Die Handlung ist eine Klasse, die eine spezielle Handlung für eine Domäne beschreibt. Die Handlungen werden durch Standards, wie LOINC oder SNOMED, benannt.

➤ **Handlungsbeziehung**

Handlungsbeziehung ist eine Klasse, die die Beziehung zwischen Handlungen beschreibt. Dies kann eine chronologische, inhaltliche und konstruktive Beziehung sein. So wird beschrieben, in welcher Abhängigkeit zum Beispiel ein Labortest zu einer Labortestanforderung steht.

Das Domain - Message Information Model (D-MIM)

Abgeleitet von den RIM werden die D-MIMs. Diese beschreiben die verwendeten Klassen innerhalb einer Domäne für einen bestimmten Zweck und stellen eine verfeinerte Teilmenge von den RIMs dar. Dies geschieht durch das Klonen der Basisklassen des RIMs; zum Beispiel wird eine Einheit für den Patienten und eine weitere für den behandelnden Arzt definiert. Die D-MIMs legen zusätzlich die

Datentypen und das Vokabular für Attribute fest und werden für die Erzeugung von domänenspezifischen Nachrichten verwendet ^[iHCI10].

Das Refined - Message Information Model (R-MIM)

Abgeleitet von den D-MIMs werden die R-MIMs. Diese beschreiben die Kategorie der Interaktionen innerhalb einer Domäne. Dabei wird das D-MIM durch das Entfernen von unnötigen Klassen und Attributen sowie das Ändern von Beziehungen verfeinert. Das Modell wird immer mehr eingeschränkt, bis es möglichst spezifisch ist. Es wird unter vier möglichen Einschränkungen (constraints) unterschieden.

- **Appearance constraints** (kommt es vor)
- **Cardinality constraints** (wenn ja, wie oft)
- **Type constraints** (wie sieht es aus)
- **Vocabulary domain constraints** (welche Werte sind zugelassen)

Eine Spezialform der R-MIM sind die Common Message Element Types (CMETs). Diese können als Bausteine in anderen D-MIMs und R-MIMs eingesetzt werden und können auch weitere CMETs enthalten.

Die Hierarchical Message Description (HMD)

Aus den R-MIM zusammen mit dem dazugehörigen RIM kann das HMD abgeleitet werden, dies ist eine tabellarische Auflistung der verwendeten Attribute und Klassen. Die Klasse mit der kleinsten Kardinalität bildet dabei das Wurzelement. Die weiteren Klassen werden in einer Baumstruktur ausgehend von der Wurzelklasse abgeleitet. Aus den HMD wird mittels der Implementation Technology Specification (ITS) das konkrete Format für die strukturierten Daten erzeugt ^[iHCI10].

Das Storyboard

Das Storyboard ist die textuelle Beschreibung eines Szenarios, zum Beispiel eines Behandlungsprozesses. Das Storyboard bildet die Grundlage für das Interaktionsmodell und die Informationsmodelle.

Das Interaktionsmodell

Das Interaktionsmodell beschreibt die Zuordnung der R-MIMs zu den Interaktionen. Es bildet daher zusammen mit den Informationsmodellen (speziell den R-MIMs) die Grundlage zur Erzeugung der Schemata. Dabei werden durch die Interaktion folgende -Eigenschaften festgelegt:

- **Receiver responsibilities** (Verpflichtungen des empfangenden Systems)
- **Trigger event** (Wie weiß das sendende System, dass eine Nachricht versendet werden muss?)
- **Message type** (Welche Struktur hat die Nachricht?)
- **Event Control Act Wrapper** (Der Typ des Triggers)
- **Sending application role** (Welches System versendet?)
- **Receiving application role** (Welches System empfängt?)

Das Message Development Framework (MDF)/HL7 Development Framework (HDF)

Der Vorgang des Generierens wird als Message Development Framework (MDF) oder auch HL7 Development Framework (HDF) bezeichnet und beschreibt den Weg der Verknüpfung der Informationsmodelle (statisch) und des Interaktionsmodells (dynamisch) zum Erzeugen der XML-Schemata. In der Abbildung VI wird der Modellierungsprozess verdeutlicht ^[HCl10].

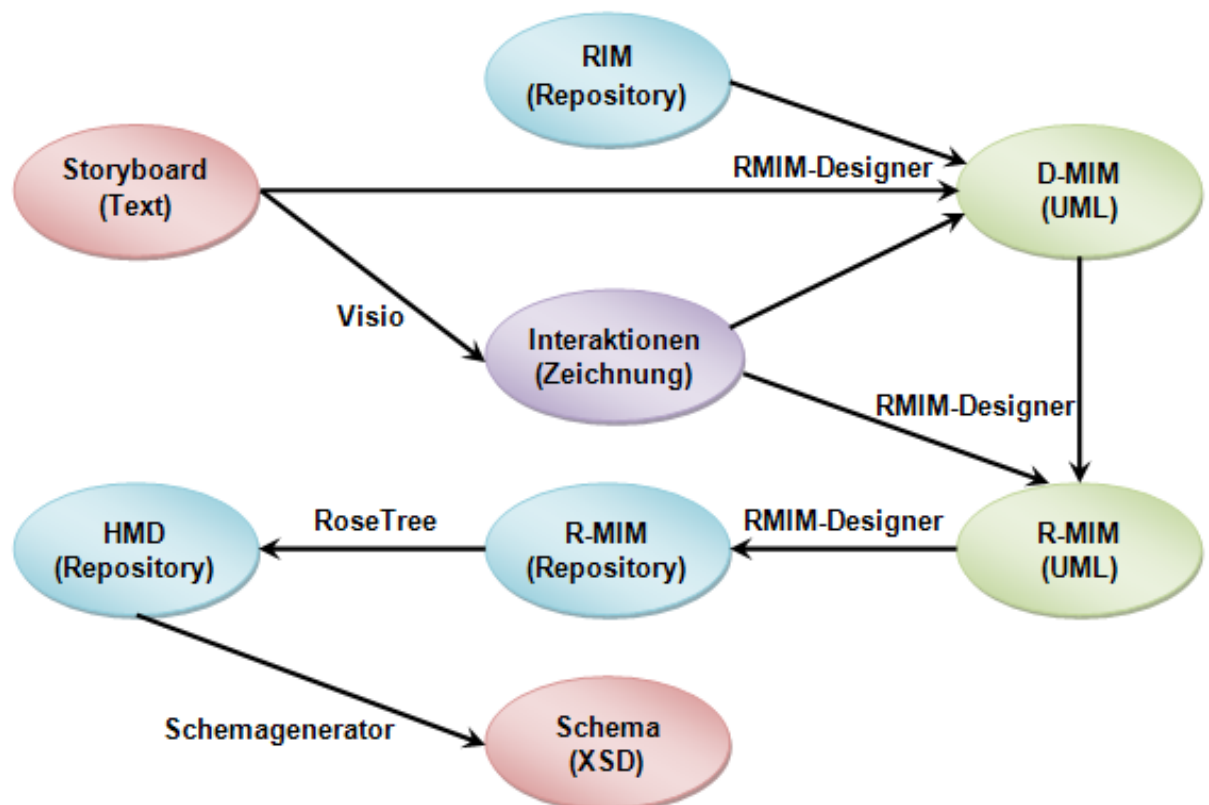


Abbildung VI Prinzip der HL7 Schemagenerierung, eigene Darstellung

Die HL7 Templates

Ein HL7 Template ist ein vom RIM abgeleiteter Satz an Einschränkungen, der es ermöglicht, die Dokumente automatisiert identifizieren, auswerten und weiterverarbeiten zu können. Die Templates werden erstellt, da es immer wiederkehrende Datenstrukturen gibt; zum Beispiel werden bei Laboranalysen immer die gleichen Arten an Daten dokumentiert.

Die HL7 Templates können sowohl den Inhalt wie auch die Struktur eines CDA Dokuments einschränken. So kann ein OP-Bericht zu zum Beispiel einem Entlassungsbrief Unterschiede aufweisen. Desweiteren können auch Einzelinformationen als erforderlich definiert werden ^[iHLTemp10].

Die HL7 Version 3 Datentypen

Da die Attribute der aus dem RIM abgeleiteten Klassen aus Name, Datentyp und Kardinalität bestehen, bilden die HL7 Datentypen mit die Basis für die Erstellung der verschiedenen HL7 Version 3-Dokumente.

Weil die Datentypen die Bedeutung der Attribute wiedergeben, können die Informationen ohne Kenntnis des Datentyps nur schwer interpretiert werden. Dabei unterscheidet man zwischen basic und generic data types. Die basic data types werden zur Definierung von zum Beispiel Kodierungen, Identifikationen, Namen, Adressen und Text verwendet. Die generic data types dagegen werden zur Definierung von zum Beispiel Datensammlungen, Listen, Intervallen und Datensets verwendet. Die Qualität der Informationen kann durch den Einsatz der generic data types erhöht werden ^[iHC110].

In der Folgenden Tabelle (Tabelle I) sind einige der HL7 Version 3 Datentypen ^[iHC110] dargestellt.

Datentyp	Bedeutung
Null	Nothing
AD	Postal And Residential Address
Any	Everything
BL	Boolean
CD	Concept Descriptor
CE	Coded with Equivalent
CS	Coded Simple Value
CV	Code Value
ED	Encoded Data
EN	Entity Name
II	Instance Identifier
ON	Organization Name
PN	Person Name Type
PQ	Physical Quantity
PQY	Quantity
REAL	Real Number
ST	Character String
TEL	Telecommunication Address
TS	Point in Time

Tabelle I Die wichtigsten HL7 Version 3-Datentypen, eigene Darstellung

Desweiteren gibt es die Möglichkeit Folgen von Datentypen ^[iHCI10] zusammenzufassen. In der folgenden Tabelle (Tabelle II) ist eine Auswahl an möglichen Folgen dargestellt.

Datentyp	Bedeutung
IVL	Interval
SET	Satz
BAG	Packet
RTO	Ratio
LIST	Sequence

Tabelle II *Zu verwendende Datentypen zum Zusammenfassen von Datentypenfolgen, eigene Darstellung*

Ein Beispiel für einen Satz an realen Zahlen wäre „SET<REAL>“.

Der Inhalt der Elemente in dem CDA-Dokument kann durch Codierungen näher beschrieben werden, so dass diese computergestützt ausgewertet werden können. Es gibt dabei keine feste Richtlinie, welche Codierungssysteme benutzt werden sollen, jedoch muss jedes Codierungssystem mit einem Object Identifier (OID) ^[iOID10] versehen und mit diesem in dem CDA-Dokument angegeben werden. So gilt für HL7 der OID „2.16.840.1.113883“, und für die Liste der bei HL7 registrierten Codierungssysteme wird daran noch eine „6“ angehängt. So hat zum Beispiel das Codierungssystem Logical Observation Identifier Names and Codes (LOINC) ^[iLOINC10] die OID „2.16.840.1.113883.6.1“.

Austauschformate

Die wichtigste Datengrundlage für die HL7 Version 3-Schemaentwicklung bildet das Repository. Dies ist eine Access-Datenbank. Sie beinhaltet die aktuelle Version des RIM. Das aktuelle Repository kann von der HL7-Webseite ^[iHea10] heruntergeladen werden.

Das notwendige Vokabular und die Basis-Datentypen werden im Model Interchange Format (MIF) gespeichert und sind im Repositoryverzeichnis mit enthalten. Das MIF wurde von der HL7-Gruppe entwickelt, um das Vokabular sowie die Datentypen speichern und austauschen zu können. Das MIF basiert auf XML, so dass die

Modelle unabhängig von variierenden Datenverzeichnissen wie das Repository zwischen verschiedenen Systemen ausgetauscht werden können.

Software zur Entwicklung eines HL7 Version 3-Schemas

Bei der Entwicklung von HL7 Version 3-Schemata kommen mehrere, von der HL7-Gruppe entwickelte, Tools zum Einsatz. Sie können von der HL7-Entwickler-Webseite^[iHL710] heruntergeladen werden. Nachfolgend sind die derzeit wichtigsten aufgeführt.

➤ **HL7-RMIM Designer**

Dies ist ein Plug-In für Visio Microsoft. Es fügt neben den für die R-beziehungsweise D-MIM-Entwicklung notwendigen Shapes die zusätzliche Menüfunktion „HL7“ ein. So können zum Beispiel die erstellten D- oder R-MIMs gegen das zugrunde liegende RIM validiert werden. Weiterhin bietet das Tool die Möglichkeit, die validierten MIMs im Repository abzuspeichern.

➤ **RoseTree**

Dieses Tool besteht aus einer GUI, mit deren Hilfe im Repository vorhandene RIMs, R-MIMs, HMDs und das Vokabular angezeigt und zum Teil bearbeitet werden können. Weiterhin bietet dieses Tool die Möglichkeit, ein HMD automatisiert aus einem R-MIM zu generieren und im Repository abzuspeichern. Darüber hinaus bietet das Tool zusätzlich die Möglichkeit, ein erstelltes HMD in einem XML-basierten Datenformat abzuspeichern, so dass der HL7 Version 3 Schema Generator dies einlesen kann.

➤ **HL7 Version 3 Schema Generator**

Dieses Tool generiert aus dem HMD ein HL7 Version 3 kompatibles XML-Schema. Dazu muss die erstellte HMD in das Input-Verzeichnis des Schemagenerators kopiert werden. Das XML-Schema wird in das Output-Verzeichnis generiert.

3.2.2 Generierung eines HL7 Version 3 Schemas anhand eines Beispielszenarios

Zum besseren Verständnis wird nachfolgend die HL7 Version 3-Schemagenerierung anhand eines Szenarios erklärt und es werden die Ergebnisse der einzelnen Schritte durch Abbildungen dargestellt ^[iRIMMod10].

Ausgangspunkt bildet ein Storyboard, welches ein Szenario beschreibt

1. Dr. Main hat einen Patienten mit Namen Max Mustermann. Diesen untersucht er am 20.12.2000, da er hohes Fieber und Symptome für eine Infektion hat.
2. Dr. Main nimmt eine Blutprobe, versieht sie mit einem Barcode und schickt sie in das Labor von Dr. Chemiker. Diese Blutprobe hat die ID 123456789.
3. Dr. Main füllt die elektronische Anforderung für ein komplettes Blutbild in seinem Praxissystem (DocSys) aus und schickt sie an das Laborsystem (LabSys).
 - Die Anforderung besteht aus:
 - Anforderungs-ID
 - Exemplar-ID
 - Name und Dienstleister-ID des Arztes
 - Name, ID, Geburtsdatum und Geschlecht des Patienten
 - Datum der Anforderung
 - Art der Anforderung: Komplettes Blutbild, codiert als LOINC-Code
 - Priorität
4. Das LabSys erhält die Anforderung und schickt eine Nachricht, dass es die Anforderung annimmt und beabsichtigt ihr nachzukommen.
5. Das Labor macht am 27.12.2000 das komplette Blutbild der Blutprobe, nachdem diese eingetroffen ist. Das Ergebnis des kompletten Blutbildes ist, dass die Anzahl der weißen Blutkörperchen bei 14,1 liegt.
6. Dr. Chemiker überprüft das Ergebnis des kompletten Blutbildes und bemerkt, dass die Anzahl der weißen Blutkörper unnormal hoch ist. Er beordert zusätzlich eine differenziertere Untersuchung zu der von Dr. Main beauftragten. Diese wird als Test #2 für die gleiche Anforderung hinzugefügt.

7. Dr. Chemiker genehmigt das Zurücksenden des Ergebnisses des kompletten Blutbildes durch LabSys an DocSys mit einem Vermerk, dass ein zusätzlicher Test eingeleitet wurde und das Ergebnis folgt.
8. Das Labor führt am 28.12.2000 die differenzierte Untersuchung der Blutprobe aus. Das Ergebnis der differenzierten Untersuchung ist, dass der Anteil der Neutrophilen bei 29% liegt.
9. Dr. Chemiker genehmigt das Zurücksenden des Ergebnisses der differenzierten Untersuchung an Dr. Main. Das ergänzende Ergebnis referenziert Original-Anforderung und -Ergebnis.
10. Die Anforderung ist nun fertig und wird an DocSys geschickt.

Daraus lassen sich folgende Informationen ableiten

➤ **Akteure:**

- Dr. Main (Anforderungsteller)
- Dr. Chemiker (Ausführer der Anforderung)
- Max Mustermann (Patient, Gegenstand der Anforderung)

➤ **Details der Anforderung:**

- (Blut)probe, Name und weitere Details des Patienten, Name und ID des untersuchenden Arztes, Details zur Anforderung, Priorität

➤ **(Original)Ergebnis der Anforderung:**

- Ergebnis der kompletten Blutprobe, Referenz zur Anforderung

➤ **(Zusatz)Ergebnis der Anforderung:**

- Ergebnis des differenzierten Tests, Referenz zur Anforderung und zum Original-Ergebnis

➤ **Trigger events:**

- Neue Anforderung
- Anforderung empfangen
- Ergebnisse übertragen
- Abschluss der Anforderung

➤ **Interaktionen:**

- Labortest anfordern
- Anforderung zum Labortest bestätigen und Labortest durchführen

- Ergebnisse der Labortests senden
- Abschluss der Anforderung senden

Erzeugen des Interaktionsmodells mit Visio

Wurden alle Informationen aus dem Storyboard zusammengetragen, wird anschließend das Interaktionsmodell erstellt. Dabei werden, wie in der Abbildung VII zu sehen, alle Interaktionen für das Szenario grafisch als UML-Diagramm dargestellt.

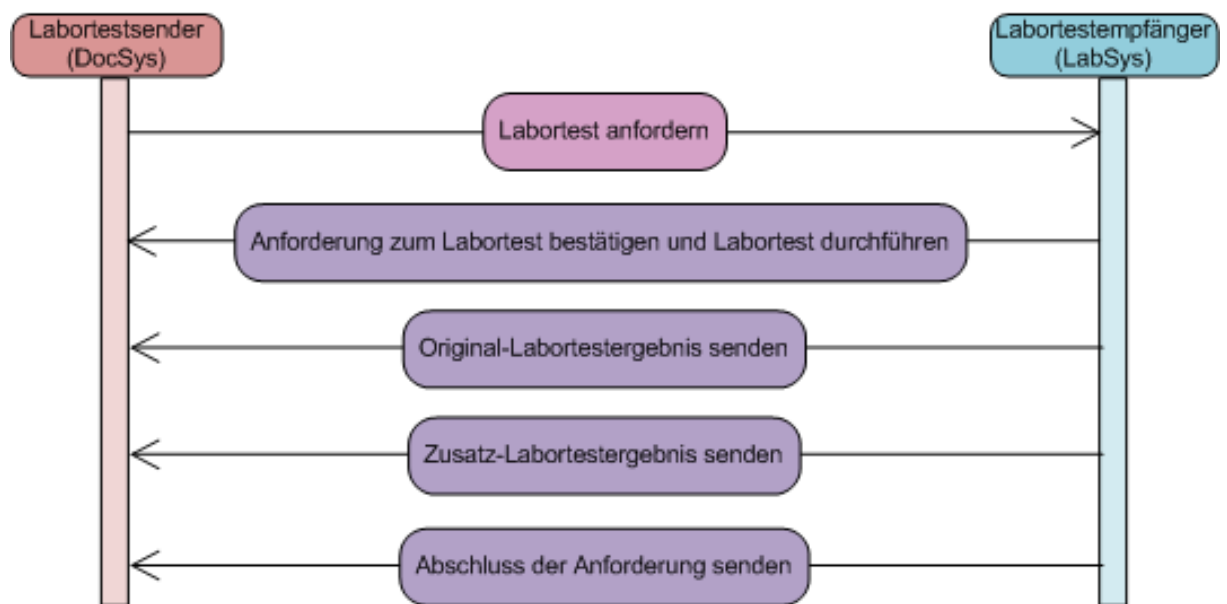


Abbildung VII Interaktionsmodell für das Beispielszenario, eigene Darstellung

Das Senden der Testergebnisse und des Abschlusses wird im weiteren Verlauf zu einer Interaktion zusammengefasst.

Erzeugen des D-MIM mittels Visio

Nun werden die gewonnenen Informationen aus dem Interaktionsmodell sowie die weiteren aus dem Storyboard gewonnenen Information im D-MIM zusammengefasst. Hierzu wird das Plug-In für Visio, der HL7-RMIM Designer, benötigt. Dabei werden alle benötigten Klassen und deren Attribute aus dem RIM geklont (zum Beispiel für den Patienten die Klasse Person mit allen benötigten Attributen). Nun werden die Klassen wie im Storyboard erkennbar zueinander in Beziehung gebracht. Dabei bilden die aus dem Interaktionsmodell abgeleiteten Interaktionen die Einstiegspunkte

für das D-MIM. Das dabei entstandene D-MIM sieht dann wie in der Abbildung VIII dargestellt aus.

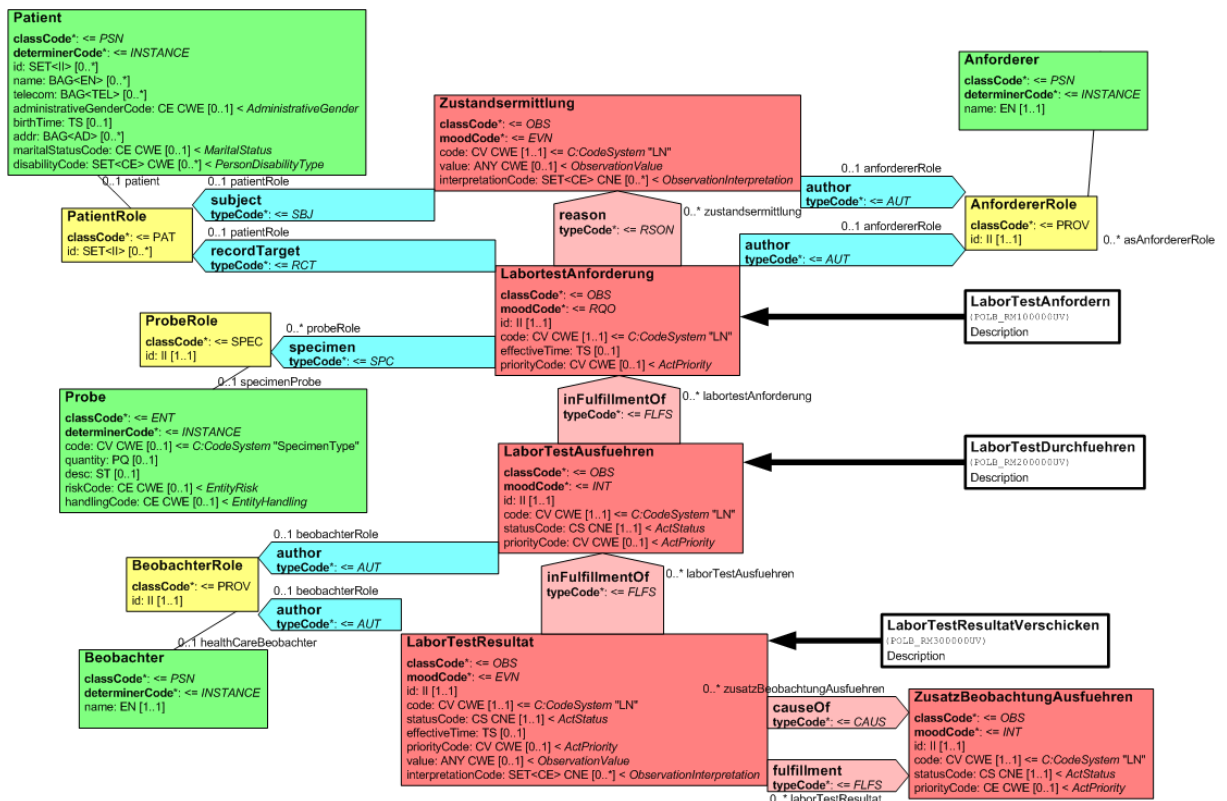


Abbildung VIII D-MIM für das Beispiel-Szenario, eigene Darstellung

Erzeugen des R-MIM mittels Visio

Im Anschluss wird für jede Interaktion ein R-MIM erstellt, dabei werden alle nicht benötigten Klassen und Attribute für die jeweilige Interaktion entfernt und gegebenenfalls weiter eingeschränkt. Für das Beispiel würden demnach drei R-MIM erstellt werden. Das R-MIM für das Verschicken der Labortestresultate ist in der Abbildung IX dargestellt.

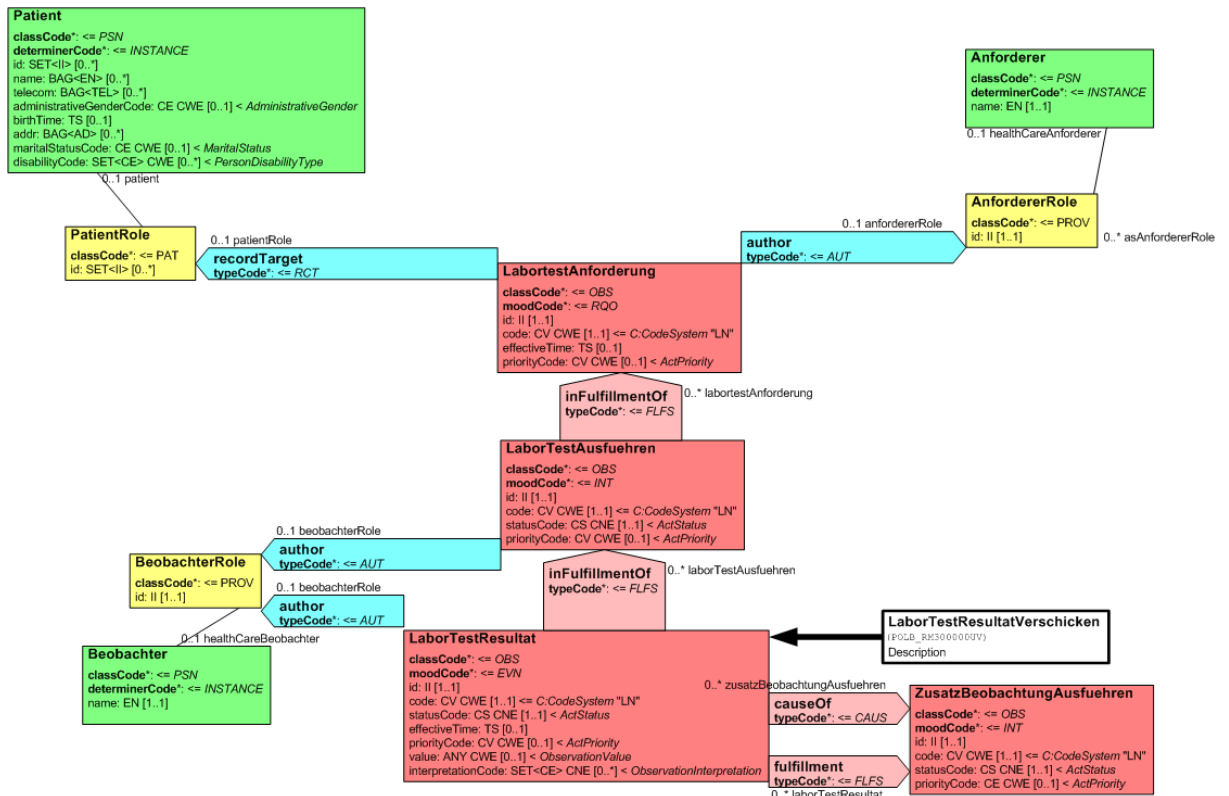


Abbildung IX R-MIM für das Verschicken des Labortestresultates, eigene Darstellung

Die anderen zwei R-MIMs können nach den gleichen Prinzip erstellt werden.

Generierung der HMD mittels RoseTree

Es werden die R-MIM Repository gespeichert und können nun mittels des Tools RoseTree geöffnet werden. Aus dem R-MIM kann im RoseTree die HMD generiert und im Repository gespeichert werden. Vorher können noch Anpassungen am R-MIM durchgeführt werden (entfernen von inaktiven Elementen zum Beispiel). In der Abbildung X wird die HMD für das Verschicken des Labortestresultates dargestellt.

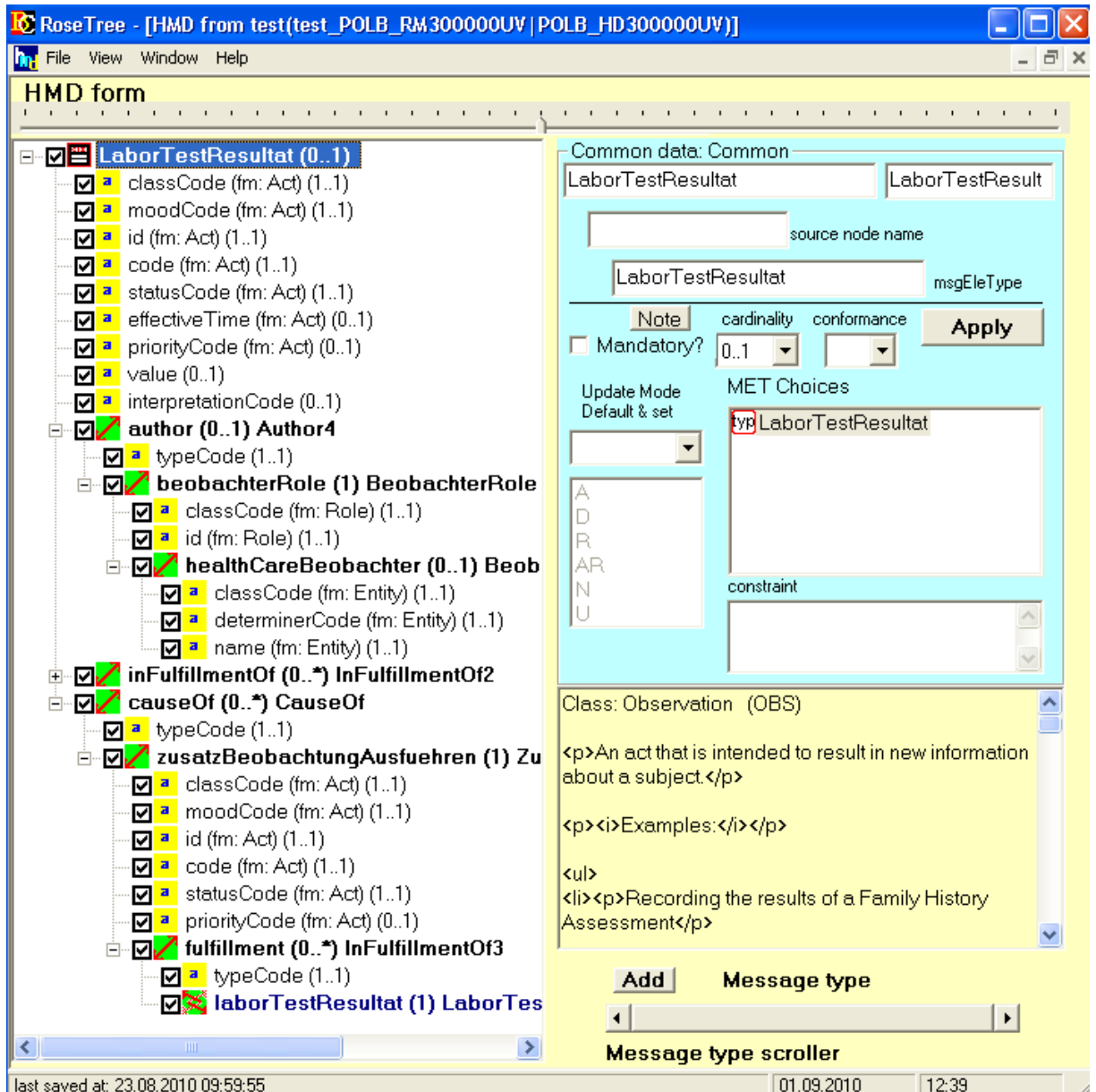


Abbildung X HMD für das Verschicken des Labortestresultates, eigene Darstellung

Erzeugen des Schemas mittels des HL7 Version 3-Schemagenerators

Abschließend wird mittels des HL7 Version 3-Schemagenerators aus der HMD das XML-Schema generiert. Da aber dabei in dem Schema kein Wurzelement definiert wird, muss das Wurzelement noch in diesem oder in einem anderen referenzierenden XML-Schema definiert werden. Ein Beispiel für ein referenzierendes Schema zeigt der Codeausschnitt II.

```
1| <?xml version="1.0" encoding="utf-8" standalone="no" ?>
2| <xs:schema targetNamespace="urn:hl7-org:v3"
3|   xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="urn:hl7-org:v3"
4|   xmlns:mif="urn:hl7-org:v3/mif" elementFormDefault="qualified">
5| <xs:include schemaLocation="POLB_HD300000UV.xsd"/>
6| <xs:element name="ClinicalDocument"
7|   type="POLB_HD300000UV.LaborTestResultat"/>
8| </xs:schema>
```

Codeausschnitt II *Definieren des Wurzelements und Referenzieren des generierten Schemas durch ein übergeordnetes Schema*

3.2.3 Die Clinical Document Architecture (CDA)

Die CDA ist ein standardisiertes Dokumentenformat, es wurde für den Austausch und die Speicherung der klinischen Daten von der HL7-Gruppe entwickelt ^[CDA10]. Ein CDA-Dokument beinhaltet sowohl medizinische Beobachtungen (zum Beispiel Diagnosen) als auch getätigte Maßnahmen (zum Beispiel Therapien). Ein CDA-Dokument kann unter anderem einen Arztbrief, eine Überweisung, einen Laborbericht oder einen Entlassbrief darstellen. Dabei gelten folgende Einschränkungen ^[Har06] für die CDA-Dokumente:

➤ **Authentizität**

Alle Daten im Dokument sind authentifizierbar, die Informationen dürfen nicht unbemerkt verändert werden und die Daten müssen einer rechtlichen Prüfung standhalten können. Desweiteren muss gewährleistet sein, dass die beabsichtigten Parteien Zugriff auf die Daten haben und die Parteien nachgewiesen werden können.

➤ **Integritätsnachweis**

Das unberechtigte Ändern durch Dritte muss erkennbar und nachgewiesen werden können.

➤ **Kompatibilität**

Das Dokument soll mit möglichst vielen Programmen kompatibel sein und es muss eine kostengünstige und einfache Implementierung gewährleistet sein. Desweiteren müssen die Daten der Dokumente unabhängig von dem zugrundeliegenden System transferiert werden können.

- **Kontext**
Ein CDA Dokument muss es ermöglichen, den Zusammenhang der Informationen zu erkennen.
- **Lesbarkeit**
Zur einfacheren menschlichen Lesbarkeit werden einfache Textdateien (kein Binärcode) verwendet, die auch für unkundige Benutzer lesbar sind.
- **Patientenorientiert**
Der bestmöglichen Behandlung des Patienten muss die größte Priorität beigemessen werden.
- **Persistenz**
CDA Dokumente müssen so lange, wie sie benötigt werden, in unveränderter Form vorliegen (zum Beispiel Archivierung).
- **Verbindlichkeit**
Die kommunizierten Daten und der Kommunikationsprozess müssen überprüft werden können und den rechtlichen Richtlinien entsprechen.
- **Vertraulichkeit**
Es muss gewährleistet sein, dass die Dokumente nur zwischen den gewollten Parteien ausgetauscht werden.
- **Verwaltung**
Ein CDA Dokument darf nur von einer Person, die für die Betreuung des Patienten verantwortlich ist, verwaltet werden und dieser zugeordnet werden können.
- **Vollständigkeit**
Das CDA-Dokument darf nicht geteilt werden, alle Daten müssen zusammenhängend sein.

Die CDA befindet sich noch in der Entwicklung. Im Jahr 2000 wurde das Release 1 und 2005 das überarbeitete Release 2 veröffentlicht. Momentan wird an dem

Release 3 gearbeitet [iRingh10]. Dieses soll auf den Erfahrungen mit der Implementierung von Release 2 aufbauen.

Der Aufbau der CDA

Die CDA-Dokumente werden, wie alle HL7 Version 3-Dokumente, aus dem HMD abgeleitet, sie beruhen auf den RIM sowie den HL7 Version 3-Datentypen. CDA-Dokumente setzen sich aus Header und Body zusammen. Im Header stehen die verwaltungstechnischen Daten, wie zum Beispiel Sender, Empfänger und Dokumententyp, im Body stehen die eigentlichen medizinischen Informationen, wie zum Beispiel Diagnosen und Therapien.

Release 1

Der Header und der Body von Release 1 sind jeweils mit eigenen Elementen („clinical_document_header“ und „body“) umschlossen. Der Codeausschnitt III zeigt die Struktur von Release 1.

```
1| <?xml version="1.0" encoding="utf-8"?>
2| <level_one>
3| <clinical_document_header>
4| <!-- Header Elemente -->
5| </clinical_document_header>
6| <body>
7| <!-- Body Elemente -->
8| </body>
9| </level_one>
```

Codeausschnitt III Die Dokumentenstruktur der Release 1

Für Release 1 wurden drei Sektion-Levels definiert.

- **Level 1:**
Nur der Header basiert auf RIM.
- **Level 2:**
Zusätzlich basiert die Sektionsebene des Bodys auf RIM.
- **Level 3:**
Auch die unterste Ebene (Datentypen und Einträge) der Sektionen basiert auf RIM.

Es wurde jedoch nur das erste Level publiziert. Dabei hat der Body nur ungenaue Strukturen, wie zum Beispiel nur einfache Tabellen, ohne semantischen Bezug zu den Daten.

Release 2

Im Gegensatz zu Release 1 wurde die Struktur bei Release 2 innerhalb des Dokumentes verändert, so sind die Headerelemente direkt als Kindselement unter dem Wurzelement („ClinicalDocument“) angeordnet. Die Bodyelemente dagegen sind durch das „component“-Element umschlossen. Der Codeausschnitt IV zeigt die Struktur von Release 2.

```
1| <?xml version="1.0" encoding="utf-8"?>
2| <ClinicalDocument>
3| <!-- Header Elemente -->
4| <component>
5| <!-- Body Elemente -->
6| </component>
7| </ClinicalDocument>
```

Codeausschnitt IV *Die Dokumentenstruktur von Release 2*

Es wird zwischen „structuredBody“ und „nonXMLBody“ unterschieden. Die Definition der Sektion-Levels gilt nur für den strukturierten Bodyabschnitt, daher wird in diesem Abschnitt nur darauf eingegangen.

Für Release 2 wurden ebenfalls drei Sektion-Levels definiert ^[ICDAR210]:

➤ **Level 1:**

Das komplette CDA-Dokument basiert auf RIM, es sind keine Template definiert. Medizinische Informationen und Daten im Body sind nur in Textform dargestellt und es sind nur ungenaue Strukturen ohne weitere Differenzierungen vorhanden (Codeausschnitt V).

```
1| <component>
2| <structuredBody>
3| <component>
4| <section>
5| <title>29.08.2005: Anamnese</title>
6| <text>Patient mit Asthma.</text>
7| </section>
8| </component>
9| </structuredBody>
10| </component>
```

Codeausschnitt V *Aufbau des Sektion-Levels 1*

➤ **Level 2:**

Bis zur Sektionsebene des Bodys sind alle Daten durch Festlegungen (Codes) definiert, es besteht die Möglichkeit dafür Templates zu definieren. Die benötigte Struktur, für zum Beispiel einen Arztbrief, ist vorgeschrieben (Codeausschnitt VI).

```
1| <component>
2| <structuredBody>
3| <component>
4| <section>
5| <code code="10164-2" codeSystemName="LOINC"
6| <codeSystem="2.16.840.1.113883.6.1"/>
7| <title>29.08.2005: Anamnese</title>
8| <text>Patient mit Asthma.</text>
9| </section>
10| </component>
11| </structuredBody>
12| </component>
```

Codeausschnitt VI *Aufbau des Sektion-Levels 2*

➤ **Level 3:**

Auch die unterste Ebene (Datentypen und Einträge) der Sektionen ist durch Codes definiert, es besteht die Möglichkeit bis zu dieser Ebene Templates zu definieren. Die benötigten Daten, für zum Beispiel einen Arztbrief, sind vorgeschrieben (Codeausschnitt VII).

```
1| <component>
2|   <structuredBody>
3|     <component>
4|       <section>
5|         <code code="10164-2" codeSystemName="LOINC"
6|           codeSystem="2.16.840.1.113883.6.1" />
7|         <title>29.08.2005: Anamnese</title>
8|         <text>
9|           Patient mit <content ID="a1">Asthma </content>.
10|        </text>
11|        <entry typeCode="COMP">
12|          <observation>
13|            <code code="195967001"
14|              codeSystem="2.16.840.1.113883.6.96"
15|              codeSystemName="SNOMED CT"
16|              displayName="Asthma">
17|              <originalText>
18|                <reference value="#a1"/>
19|              </originalText>
20|            </code>
21|          </observation>
22|        </entry>
23|      </section>
24|    </component>
25|  </structuredBody>
26| </component>
```

Codeausschnitt VII *Aufbau des Sektion-Levels 3*

Nachfolgend wird nur auf Release 2 eingegangen, da sie der derzeitige Standard für die CDA-Dokumente ist.

Der CDA-Header

Im Header sind die Informationen über den Patienten, die beteiligten Organisationen, weitere beteiligte Personen, das Dokument selbst, andere im Zusammenhang stehende Dokumente und die zeitlichen Ereignisse zusammengefasst. Die Informationen im Header ermöglichen einen Austausch des Dokuments auch über Institutsgrenzen hinweg sowie eine Zusammenführung verschiedener Dokumente zu einer Patientenakte. Den Header kann man in drei Abschnitte einteilen. Nachfolgend werden diese sowie die bedeutendsten Elemente und deren wichtigsten Attribute näher erläutert.

Eigenschaften des Dokuments:

In diesem Abschnitt werden das Dokument näher spezifiziert und die verwaltungstechnischen Informationen zu dem Dokument beschrieben.

- **code**

Dieses Element typisiert die Dokumentenart, diese wird durch das „code“-Attribut bereitgestellt. Mit dem Attribut „codeSystem“ wird das zugrunde liegende Codierungsverzeichnis bezeichnet.
- **confidentialityCode**

Mit dem „confidentialityCode“-Element wird die Vertraulichkeit des Dokuments definiert, es wird zwischen normal(N), restricted(R) und very restricted (V) im „code“-Attribut unterschieden. Das „codeSystem“-Attribut bezeichnet das Codierungsverzeichnis.
- **effectiveTime**

Das „effectiveTime“-Attribut stellt das Erstellungsdatum der Dokumentation dar.
- **id**

Das Element „id“ weist dem Dokument einen eindeutigen Identifier zu. Das „root“-Attribut dokumentiert die OID des Erstellers des Dokuments und das „extension“-Attribut den eigentlichen Identifier.
- **languageCode**

Im „code“-Attribut des „languageCode“-Elements wird die Sprache des Dokuments angegeben.
- **setId und versionNumber**

Diese Elemente dienen zur Versionierung der Dokumentation, dabei wird zwischen Hinzufügen und Ersetzen unterschieden. Beim Hinzufügen bleibt die „setId“ bestehen, die „versionNumber“ dagegen wird für jedes Dokument um 1 erhöht. Beim Ersetzen wird eine neue „setId“ und die „versionNumber“ wieder auf 1 gesetzt. Mit der Versionierung ist es möglich, mehrere CDA-Dokumente zu einer Patientenakte zusammenzuführen.
- **typeId**

In diesem Element wird das zugrunde liegende R-MIM aufgeführt. Das „root“-Attribut dokumentiert die feste OID „2.16.840.1.113883.1.3“, diese steht für alle bei HL7 registrierten Modelle. Das „extension“-Attribut steht für den eigentlichen Namen des R-MIMs, für CDA Release 2 ist dies der Identifier „POCD_HD000040“.

Beteiligten Parteien

Dabei ist zu beachten, dass alle Personen mit einem Namen zu versehen sind, desweiteren sollte jede Person mit einer Adress- und Telekommunikationsinformation versehen werden. Bei menschlichen Autoren und bei Organisationen ist die Angabe der Kommunikationsinformationen Pflicht, falls nicht bekannt muss dies mit einem „nullFlavor“-Attribut gekennzeichnet werden [iLeitfAr10].

➤ **author**

Dieses Element stellt den Autor der Dokumentation dar, dies können Personen (Heilberufler) oder auch Geräte sein.

➤ **authorization**

In diesem Element kann dokumentiert werden, inwiefern der Patient dem Verfügbar machen an Dritte zugestimmt hat.

➤ **componentOf**

In diesem Element wird der Rahmen des Aufenthalts dargestellt. Es wird unter anderem die Aufenthaltsart aufgeführt. Dies kann ein stationärer (IMP), ein ambulanter (AMP) oder ein virtueller (VR) Aufenthalt sein.

➤ **custodian**

Hier wird die Organisation, die für die Verwaltung des Dokuments verantwortlich ist, hinterlegt.

➤ **informationRecipient**

Dieses Element beschreibt den beabsichtigten Empfänger des Dokuments. Mit dem „typeCode“-Attribut wird gekennzeichnet, wer der Primäre Empfänger

des Dokuments ist (PRCP) und wer gegebenenfalls der Empfänger der Kopie ist (TRC).

➤ **authenticator**

Hier wird der verantwortliche Unterzeichner dokumentiert, zusätzlich in dem abgeleiteten Elementen „time“ und „signatureCode“ wird der Zertifizierungszeitpunkt sowie die Art der Signatur typisiert. Ist die Beglaubigung rechtskräftig, wird statt dem „authenticator“- das „legalAuthenticator“-Element verwendet.

➤ **participant**

Im „participant“-Element können die Beziehungen des Patienten zu anderen Personen oder Organisationen beschrieben werden. Dies können unter anderem Angehörige, Notfall-Kontakte und Versicherungen sein.

➤ **recordTarget**

In diesem Element sind der Patient und dessen administrative Daten hinterlegt.

Beziehungen zu vorhergehenden Dokumenten

➤ **relatedDocument**

Die Beziehung zu vorhergehenden Dokumenten kann durch das „relatedDocument“-Element wieder gegeben werden. Es kann im „typeCode“-Attribut zwischen ersetzen (RPLC), hinzufügen (APND) und transformieren (XFRM) unterschieden werden. RPLC bedeutet, dass das alte Dokument durch das aktuelle komplett ersetzt wird und die Versionsnummer beim aktuellen Dokument um 1 erhöht wird. Vom APND spricht man, wenn zu dem vorhergehenden Dokument das aktuelle (zum Beispiel eine Anlage) mit eigenem Identifier hinzugefügt wird. Wenn ein altes Dokument in das aktuelle transformiert wurde wird dies mit XFRM gekennzeichnet.

Der CDA-Body

Im Gegensatz zum Header, in dem nur die administrativen Daten aufgeführt sind, stehen die eigentlichen medizinischen Daten im Body. Wie schon auf Seite 28 angedeutet muss der Body durch ein „structuredBody“-Element oder ein „nonXMLBody“-Element umschlossen werden. Falls die Daten in einem externen Dokument (zum Beispiel in einer PDF-Datei) vorliegen, können diese durch das „nonXMLBody“-Element direkt eingebunden werden. Dies sowie die von XML unterstützte Einbindung von Binärdaten, die Binary Large Objects (BLOBs), werden durch den VHitG-Standard nicht unterstützt.

Unter dem „structuredBody“-Element untergeordnet befindet sich der richtige VHitG-konforme strukturierte Body. Dieser besteht aus mindestens einem Abschnitt (section). Falls notwendig können die Abschnitte auch ineinander verschachtelt sein. Dabei muss jeder Abschnitt lesbaren (narrativen) Text enthalten, so dass die Informationen durch den Menschen gelesen werden können. Zusätzlich muss jede Sektion mit einem Code, der über die Bedeutung der Sektion Auskunft gibt, versehen werden (siehe Sektion-Level 2). Die Darstellung des Textes kann durch Möglichkeiten der Form- und Strukturgebung, fett, kursiv und unterstrichen, um nur ein paar wenige zu nennen, beeinflusst werden. Diese werden wie bei HTML durch Kürzel, zum Beispiel „
“ für Zeilenumbruch, dargestellt. Durch die „content“-Elemente im Text können bestimmte Informationen gekennzeichnet werden und dadurch auf CDA Einträge (entries) verweisen (siehe Sektion-Level 3) ^[iLeitfAr10].

Die CDA Einträge stellen den computerlesbaren Teil des Bodys dar. In diesem Abschnitt werden die markierten Informationen innerhalb des Textes durch Codierungen genauer und für den Computer auswertbar spezifiziert. Dies können unter anderem Befunde, Medikationen und auch Verweise auf externe Daten sein. Besonders in diesem Teil des CDA-Dokuments ist die objektorientierte Idee deutlich zu erkennen. Für die verschiedenen Typen der Inhalte sind unterschiedliche übergeordnete Elemente vorhanden. So werden Beobachtungen, wie zum Beispiel eine Diagnose oder ein Befund, durch das „observation“-Element definiert. Prozeduren, wie zum Beispiel diagnostische Eingriffe oder Operation, dagegen

werden durch das „procedure“-Element definiert. Selbstverständlich weisen beide Elemente unterschiedliche Strukturen und Inhalte auf. In der Abbildung XI werden der Aufbau und die in diesem Kapitel beschriebenen Inhalte eines CDA-Dokuments dargestellt.

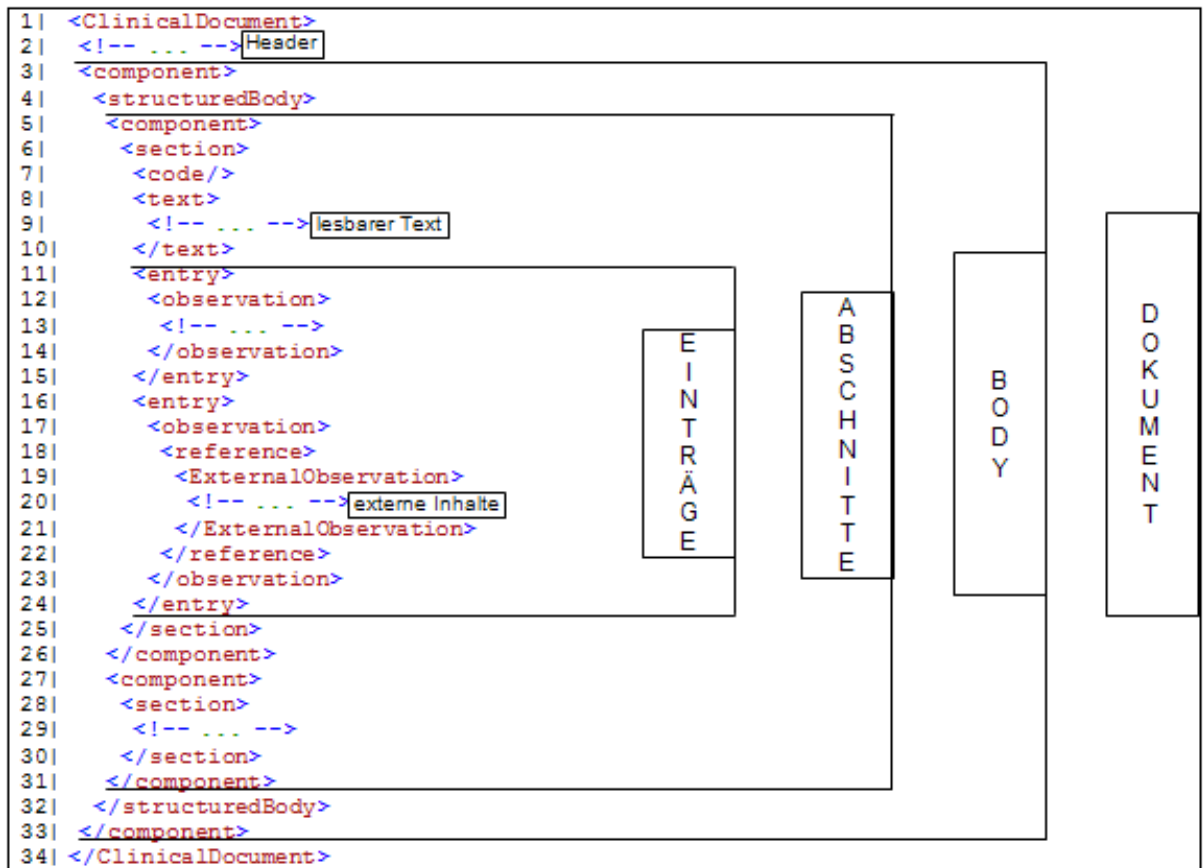


Abbildung XI XML-Struktur eines CDA-Dokuments, eigene Darstellung

4 Transformation einer HL7 Version 2-Nachricht in ein VHitG-Standardkonformes HL7 Version 3-Dokument

Für die Transformation einer HL7 2.x-Nachricht in ein HL7 Version 3-Dokument wurde eigens eine neue Produktion auf der Integrationsplattform Ensemble angelegt. Eine Produktion in Ensemble besteht aus Business Services, Prozesse und Operationen. Ein Service definiert die Aufnahme, ein Prozess die Verarbeitung und eine Operation die Ausgabe der Daten. In den Services und den Operationen wird mittels Adapterklassen das Abbilden der Daten in Ensemble gesteuert. Im Business Prozess können Regeln zur Verarbeitung der Daten in Ensemble angelegt werden. Dies können zum Beispiel Transformations- oder Weiterleitungsregeln sein.

4.1 Anlegen der Produktion im Ensemble Management Portal

Um den Transformationsprozess zu gewährleisten, wird im Ensemble Management Portal eine neue Produktion angelegt. Dazu wird beim Anlegen der Produktion als Typ „HL7-Messaging“ ausgewählt. Dabei wird eine HL7 Version 2 Routing Produktion erstellt. Es werden ein Business Service, zwei Business Prozesse und vier Business Operationen wie in der Abbildung XII zu sehen automatisch angelegt.

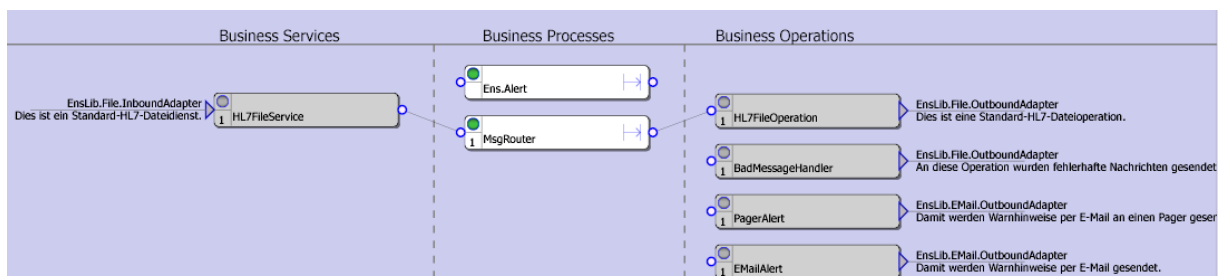


Abbildung XII Eine Standard HL7-Produktion

Der Business Service beinhaltet einen HL7 Version 2 Datei Service, der für das Einlesen einer HL7 Version 2-Datei verantwortlich ist. Die zwei Business Prozesse

sind einerseits ein Nachrichten Router und andererseits ein Fehler Router. Der Nachrichten Router ist für das Routen von einem Business Service zu einer Business Operation verantwortlich. Der Fehlerrouter wiederum leitet aufgetretene Fehler an eine Business Operation weiter. Bei den Business Operationen handelt es sich um einen Datei Service zum Abspeichern einer HL7 Version 2-Datei und drei Operationen für die Fehlerbehandlung. Zwei der Business Operationen zur Fehlerbehandlung können jedoch entfernt werden, da sie nur dem Zweck dienen, eine E-Mail an einen Pager beziehungsweise an eine E-Mail-Postfach mit der Fehlerbeschreibung zu versenden. Da der Business Prozess zur Fehlerbehandlung für das Erstellen dieser E-Mail verantwortlich ist, kann er genauso mit vernachlässigt und daher auch mit entfernt werden. Nun müssen der Routingprozess und die Dateioperation zum Abspeichern der Datei geändert werden, da beide auf die Durchleitung einer HL7 Version 2-Nachricht konzipiert sind, aber durch den Transformationsprozess ein HL7 Version 3-Dokument entstehen soll. Sowohl für den Routingprozess als auch für die Speicherdateioperation müssen die zugrundeliegenden Klassen angepasst werden. In der Abbildung XIII wird diese Produktion dargestellt.

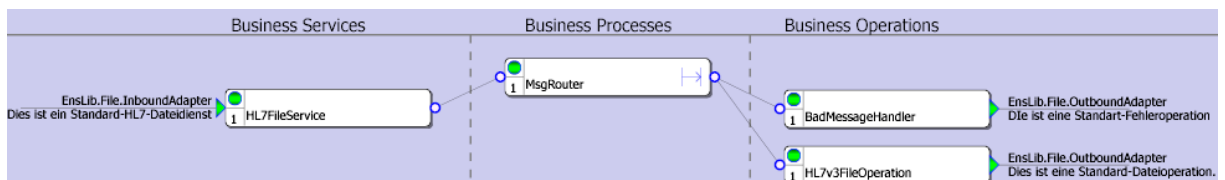


Abbildung XIII *Übersicht über die Produktion zur Transformation einer HL7 Version 2-Nachricht in ein HL7 Version 3-Dokument, eigene Darstellung*

Desweiteren muss bei dem Routingprozess über den Regeleditor die eigentliche Transformation mit eingebunden werden.

4.2 Erstellung der Transformationsklasse im Studio

4.2.1 Prinzip der Transformation

Wie in dem Abschnitt 3.2 ab Seite 10 beschrieben weist zum Beispiel der Header eines HL7 Version 3-Dokuments im Gegensatz zu einer HL7 Version 2-Nachricht neben den administrativen Angaben zu dem Dokument selber auch Informationen zu dem betroffenen Patienten auf. Auch ist die Anordnung der Daten in einer anderen Reihenfolge wie in einer HL7 Version 2-Nachricht. Daher ist es von sehr großem Vorteil, vor der eigentlichen Transformation die HL7 Version 2-Nachricht in eine HL7 Version 2.xml-Nachricht zu übertragen, da nun die Elemente einfach unter Verwendung von XPath⁴ ausgelesen werden können.

In der Transformation werden die Elemente des HL7 Version 3-Dokuments durch das Abfragen und gegebenenfalls Transformieren des Inhalts der Elemente der HL7 Version 2.xml-Nachricht übertragen.

4.2.2 Data Transformation Language (DTL)

Der HL7 Version 2 Datei Service nutzt die dazu gehörige Nachrichtenklasse `EnsLib.HL7.Message`, die die Felder einer HL7 Version 2-Nachricht in die HL7 Version 2.xml-Struktur bringt. So können die Daten innerhalb der Transformationsklasse einzeln abgefragt werden. Durch die Verwendung der Nachrichtenklasse kann eine Transformation zweier HL7 Version 2-Nachrichten mit Hilfe der Extensible Stylesheet Language Transformation (XSLT)⁵-ähnlichen Data Transformation Language (DTL) Syntax erstellt werden. Die DTL stellt dabei die

⁴ XPath ist eine Abfragesprache für XML, mit deren Hilfe Elemente in einem XML-Dokument adressiert und somit einfach auslesbar gemacht werden.

⁵ XSLT ist eine Sprache zur Transformation eines XML-basierten Dokuments, es ist Teil der Extensible Stylesheet Language (XSL).

Westfälische Hochschule Zwickau

Transformation, wie in der Abbildung XIV gezeigt, dar und wird zusätzlich durch den Visualeditor graphisch unterstützt. Die Klasse ist vom Typ Ens.DataTransformDTL.

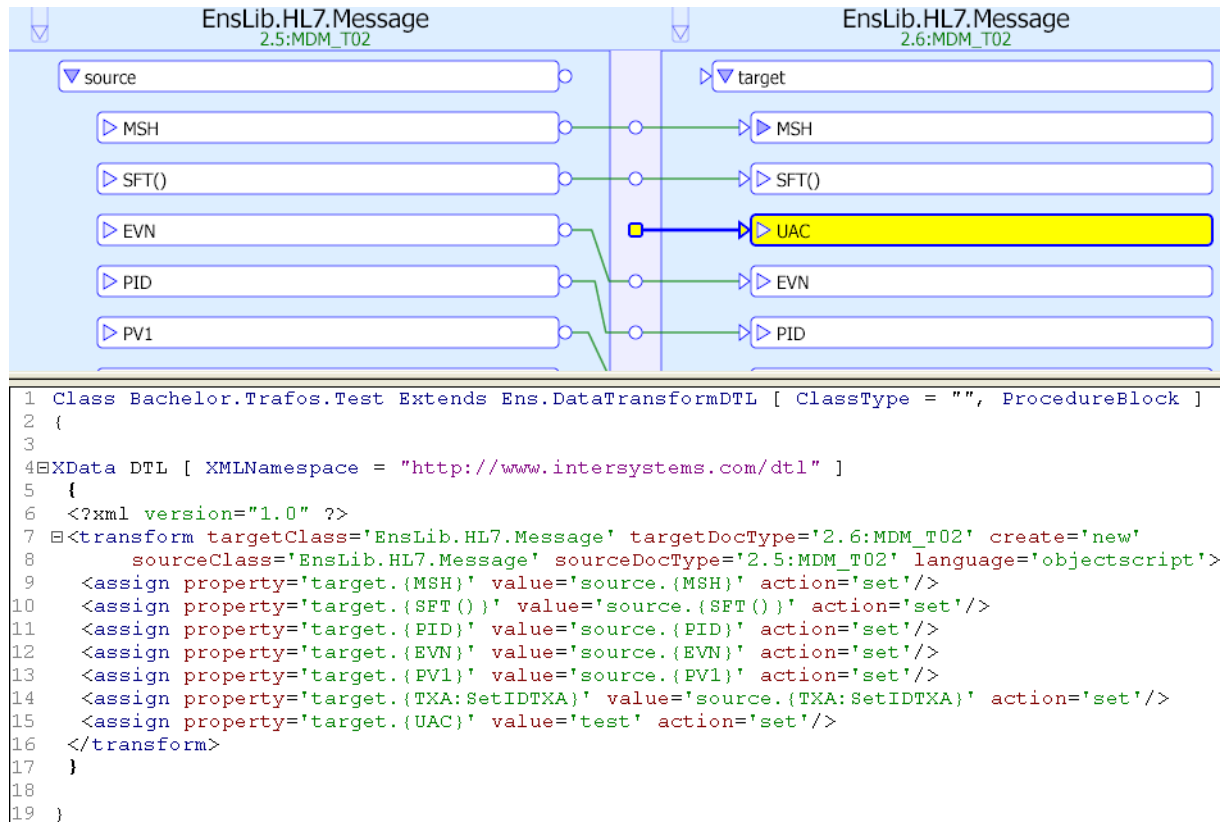


Abbildung XIV Prinzip der Data Transformation Language mit Nutzung des
Visualeditors, eigene Darstellung

So ist es möglich, die Transformation durch einfaches Drag'n'Drop zu erstellen. Parallel wird dazu der Quellcode in der DTL generiert. Es besteht ebenso die Möglichkeit, kompliziertere Transformationsregeln direkt in der DTL zu schreiben. Die Nutzung der DTL setzt jedoch voraus, dass sowohl für das Ausgangs- als auch für das Zielobjekt eine Klasse definiert ist, die den Aufbau des jeweiligen Objektes beschreibt. In der Abbildung XIV sind das Quell- und das Zielobjekt vom Typ HL7 Version 2, und die Klasse EnsLib.HL7.Message beschreibt den Aufbau des HL7 Version 2-Objektes. Zusätzlich muss für diesen Objekttyp noch der sogenannte DocType definiert werden. Dieser gibt die genaue HL7 Version und das Trigger Event für das Quell- und das Zielobjekt an. Wie in der Abbildung XIV zu sehen, ist das Quellobjekt vom Typ Version 2.5, es beschreibt das Trigger Event T02, das

Westsächsische Hochschule Zwickau

Zielobjekt dagegen ist vom Typ Version 2.6 und beschreibt ebenfalls das Trigger Event T02. Wie im Visualeditor in der Abbildung XIV erkennbar, ist das Segment UAC (User Authentication Credential) in den älteren Versionen nicht vorgesehen ist und daher wie in diesem Fall zusätzlich erzeugt werden muss.

Für die Transformation mit Hilfe der DTL ist es zwingend erforderlich, dass eine Nachrichtenklasse, die die Struktur ähnlich eines XML Schemas (XSD)⁶ der HL7 Version 3-Dokumente beschreibt, zu erstellen ist. Diese Klasse wurde in der Einleitung unter dem Punkt Zielsetzung auf Seite 2 als interne XML-Schemadefinition zum Verarbeiten von HL7 Version 3-Dokumenten in Ensemble bezeichnet. Im Laufe der Arbeit mit Ensemble und dem Auseinandersetzen mit HL7 Version 3 wurde erkennbar, dass die Erzeugung dieser Klasse innerhalb dieser Bachelorarbeit nicht durchführbar ist. Aus diesem Grund wurde beschlossen, die Transformation für das Beispielszenario unabhängig von der Erstellung der Nachrichtenklasse für HL7 Version 3-Dokumente zu erzeugen. Es wird nur eine Nachrichtenklasse ohne jegliche Strukturdefinitionen wie im Codeausschnitt VIII dargestellt verwendet.

```
1| Class Bachelor.Message Extends (%Persistent, %XML.Adaptor) [ClassType=persistent,  
  | Inheritance=right, ProcedureBlock]  
2| {  
3|   Property Content As %GlobalCharacterStream;  
4|   Property Source As %String(MAXLEN=250, TRUNCATE=1);  
5| }
```

Codeausschnitt VIII *Inhalt der HL7 Version 3-Nachrichtenklasse*

Dabei wird in dem Property „Content“ die XML-Struktur hinterlegt und im Property „Source“ wird der Name des Dokuments gespeichert.

Weil die XML-Struktur erst während der laufenden Transformation erzeugt wird, kann die DTL nicht genutzt werden; diese benötigt die Struktur schon zum Erzeugen der

⁶ XSD ist das XML Schema, um den Aufbau einer XML-Datei zu beschreiben, es hat die Document Type Definition (DTD) abgelöst.

Westsächsische Hochschule Zwickau

Transformation. Aus diesem Grund muss für die Transformation eine Klasse vom Typ `Ens.DataTransform` erzeugt werden.

4.2.3 Aufbau der Klasse zur Transformation

Die Transformationsklasse ist, wie im vorhergehenden Abschnitt beschrieben vom Typ `Ens.DataTransform` und muss daher in der Caché ObjectScript-Syntax geschrieben werden. Im Gegensatz zu der Verwendung der DTL, wo die nötigen Elemente automatisch erzeugt werden, müssen die Elemente, die der HL7 Version 3-Struktur entsprechen, bei jeder Transformation neu erzeugt werden. Da bei XML jedes Element am Anfang ein öffnendes, am Ende ein schließendes Tag besitzt und der Elementinhalt von diesen beiden Tags umschlossen ist, muss dies in der Caché ObjectScript-Syntax ebenfalls so gestaltet werden. Das heißt, das Prinzip der Erzeugung eines Elements ist immer:

- **Öffnendes Tag schreiben**
- **Attribute und/oder Elementinhalt schreiben**
- **Schließendes Tag schreiben**

Der Codeausschnitt IX stellt dieses Vorgehen in der Caché ObjectScript-Syntax dar.

```
1| //1. Element öffnen
2| s sc=writer.Element("1. Element")
3| //2. Element öffnen
4| s sc=writer.Element("2. Element")
5| //1. Attribut schreiben
6| s sc=writer.WriteAttribute("1. Attributs", "Inhalt des 1. Attributs")
7| //2. Attribut schreiben
8| s sc=writer.WriteAttribute("1. Attributs", "Inhalt des 1. Attributs")
9| //Inhalt des 2. Elements schreiben
10| s sc=writer.Write("Inhalt 2. Element")
11| //2. Element schließen
12| s sc=writer.EndElement()
13| //1. Element schließen
14| s sc=writer.EndElement()
```

Codeausschnitt IX *Syntax der XML-Dokument Erzeugung*

Wie im Codeausschnitt IX angedeutet geht mit zunehmender Komplexität die Übersichtlichkeit immer mehr verloren und die Anzahl des Quellcodes steigt dadurch ebenfalls sehr stark an.

Westsächsische Hochschule Zwickau

Die implementierte Transformation befolgt das auf Seite 38 beschriebene Prinzip. Die XML-Struktur wird dabei HL7-konform angelegt. In der Anlage I ab Seite XVI ist der Quellcode für die Transformationsklasse dargestellt.

4.2.4 Workflow der Datentransformation

Die Abbildung XV zeigt den Workflow der Transformation einer HL7 Version 2-Nachricht in ein VHitG-Standardkonformes HL7 Version 3-Dokument.

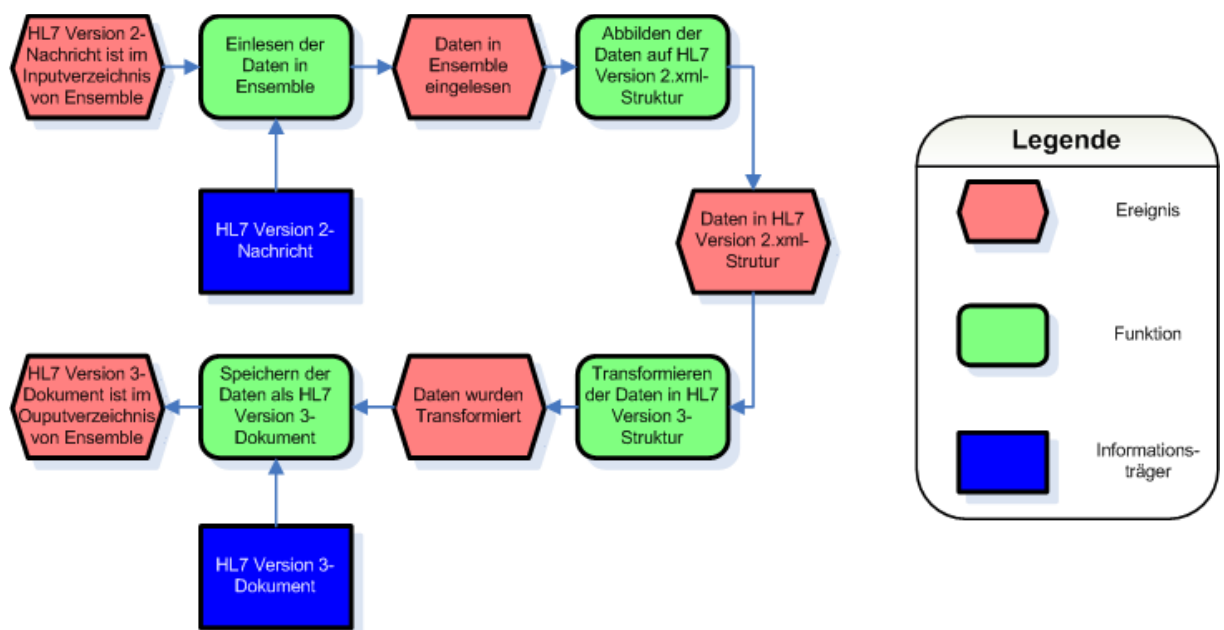


Abbildung XV Workflow für die Transformation, eigene Darstellung

5 Diskussion und Ausblick

5.1 Zielerreichung

Ein Ziel dieser Bachelorarbeit war das Beschreiben der Unterschiede zwischen HL7 Version 2-Nachrichten und HL7 Version 3-Dokumenten. Dies wurde durch das Erläutern des Aufbaus und Prinzips beider Versionen erreicht.

In dieser Bachelorarbeit wurde auch der Zusammenhang zwischen HL7 Version 3, der CDA und dem VHitG-Arztbrief näher erklärt.

Das Ziel zur Erstellung einer internen Schemadefinition in der Integrationsplattform Ensemble konnte nicht erreicht werden. Aus diesem Grund wurde die Generierung von HL7 Version 3 XML-Schemata anhand eines Beispiels ausführlich beschrieben.

Ein weiteres Ziel dieser Bachelorarbeit war das Erstellen einer Transformation von einer HL7 Version 2-Nachricht in ein VHitG-Standardkonformes HL7 Version 3-Dokument. Diese Transformation wurde für eine Beispielnachricht erstellt und überführt diese Nachricht in ein HL7 Version 3 konformes Dokument. Hierfür wurde auch ein Workflow, der diese Verarbeitung in der Integrationsplattform Ensemble beschreibt, erstellt.

5.2 Diskussion

Das Ziel die Erstellung eines internen Schemas als Caché ObjectScript-Klasse in Ensemble konnte aufgrund der Komplexität dieser Klasse nicht realisiert werden da zudem auch anfangs die Grundlagen um HL7 Version 3 fehlten und erst im Lauf der Arbeit erarbeitet werden mussten. Eine weitere Problematik stellten die zum Teil älteren Informationsquellen dar. Auch konnte manch Informationsmaterial nicht erreicht werden, da der Zugang eine kostenpflichtige Mitgliedschaft in einer der HL7-Gruppen voraussetzte. Aus diesem Grund musste auf zum Teil sich widersprechenden Informationsquellen zurückgegriffen werden, die zu einer Verzögerung in der Recherche sorgten. Eine weitere Verzögerung bei der Anfertigung der Bachelorarbeit stellte der Ausfall des Netzlaufwerks an der

Hochschule dar, da zu diesem Zeitpunkt noch alle Daten auf diesen Netzlaufwerk gespeichert waren und die Wiederherstellung der Daten fast eine Woche in Anspruch nahm.

5.3 Ausblick

Diese Arbeit bietet die Basis, um sich einen Überblick über die Grundkenntnisse von HL7 Version 3 zu verschaffen und kann daher für weitere Entwicklungen um HL7 Version 3 verwendet werden. Diese Arbeit bietet zudem Grundlagen für weitere Transformationen und da die Arbeit das Fundament für das Verständnis um HL7 Version 3 liefert auch für das Erstellen des internen Schemas für HL7 Version 3-Dokumente in Ensemble. Desweiteren können die Erklärungen zum Erstellen von HL7 Version 3-Schemata verwendet werden um den Objektorientierten Aufbau von HL7 Version 3 zu verstehen. Das vorhandene Beispiel einer HL7 Version 3-Schemaerzeugung kann zusätzlich verwendet werden um derartige Schemata eigenständig zu erzeugen.

6 Thesen

1. Für eine schnelle und unabhängige Kommunikation im Intersektoralen Bereich ist ein gemeinsames Datenformat notwendig, dieses Format bietet die HL7 Version 3.
2. Auch in naher Zukunft wird die Kommunikation im stationären Bereich auf Basis der HL7 Version 2 geschehen und daher müssen die Daten, um die Kommunikation mit dem ambulanten Bereich zu gewährleisten, in das HL7 Version 3-Format transformiert werden.
3. Für eine zuverlässige Transformation einer HL7 Version 2-Nachricht in ein HL7 Version 3-Dokument sollte die Nachricht erst in die HL7 Version 2.xml-Struktur überführt werden.
4. Da die HL7 Version 2 keinen Freitext mit Unterstützung durch Codierungen wie bei der HL7 Version 3 vorsieht ist die Transformation nur bedingt möglich.
5. Ein VHitG-konformes HL7 Version 3-Dokument kann nur dann erstellt werden wenn die notwendigen Felder in der HL7 Version 2-Nachricht gesetzt sind.
6. HL7 Version 3 wird in Zukunft eine immer größere Rolle im Intersektoralen Bereich spielen.

Quellenverzeichnis

- [Har06].....**Hartmut Bärwolff, Frank Victor, Volker Hüsken.** *IT-Systeme in der Medizin: IT-Entscheidungshilfe für den Medizinbereich; Konzepte, Standards und optimierte Prozesse.* Wiesbaden : Friedrich Vieweg & Sohn Verlag | GWV Fachverlage GmbH, 2006.
- [iBer07].....**Bernd Blobel, Frank Oemig.** *HL7/Sciphox Spezifikationen in Kooperation mit VHitG und KBV für die deutsche eHealth - Plattform.* [Online] 2007. [Abgerufen am 21. September 2010]
http://www.eh-cc.de/wp-content/html/agdgi/down/KIS-2007_07_Blobel.pdf.
- [iCDA10].....**HL7 International.** *Clinical Document Architecture.* [Online]
[Abgerufen am 14. September 2010]
<http://www.hl7.org/implement/standards/cda.cfm>.
- [iCDAR210].....**Dolin, Robert H., et al.** *HL7 Clinical Document Architecture, Release 2.0.* [Online] [Abgerufen am 12. September 2010]
<http://xml.coverpages.org/CDA-Release2-Unofficial.html>.
- [iCorHe10].....**CorePoint Health.** *The HL7 Evolution.* [Online] [Abgerufen am 05. September 2010]
<http://www.corepointhealth.com/sites/default/files/whitepapers/hl7-v2-v3-evolution.pdf>.
- [iEnsem10].....**InterSystems Corporation.** *InterSystems Ensemble.* [Online] [Abgerufen am 20. April 2010]
<http://www.intersystems.de/ensemble/index.html>.
- [iFunk06].....**Funkat, Gert, Smers, Stefan und Winter, Alfred.** *51. Jahrestagung der Deutschen Gesellschaft für Medizinische Informatik, Biometrie und Epidemiologie: Sicherer Austausch medizinischer Text-, Bild- und Abrechnungsdaten in der intersektoralen Versorgung am*

- Beispiel des Universitätsklinikums Leipzig AöR.* [Online] 1. September 2006. [Abgerufen am 07. April 2010]
<http://www.egms.de/static/de/meetings/gmds2006/06gmds065.shtml>.
- [iHBA10].....**Healthbase Australia.** *Workflow in Healthcare - Extract from HL7 RIM.* [Online] [Abgerufen am 11. September 2010]
http://workflow.healthbase.info/monographs/RIM_rationale.html.
- [iHCI10].....**Healthcareinformatics.** *The HL7V3.0 Primer.* [Online] [Abgerufen am 20. September 2010]
<http://healthcareinformatics3000feet.blogspot.com/2008/01/hl7v30-primer.html>.
- [iHea10].....**Health Level Seven International.** *Downloadseite für RIMs.* [Online] [Abgerufen am 10. September 2010]
<http://www.hl7.org/implement/standards/rim.cfm>.
- [iHeal10].....**Health Level Seven (HL7).** *FuE-Bereich IuK-Systeme im Gesundheitswesen IG.* [Online] [Abgerufen am 18. September 2010] http://www.mh-hannover.de/fileadmin/zentrale_einrichtungen/tumorzentrum/Klinisches_Krebsregister/Arbeitsgruppe_TuDaSch/Dokumentensammlung/weitere_Themen/KRG_AGS_DOK_HL7200302.pdf.
- [iHL710].....**HL7-Gruppe.** *Entwicklerseite der HL7-Gruppe.* [Online] [Abgerufen am 10. September 2010] <http://gforge.hl7.org/gf/>.
- [iHL7Stan10].....**HL7 Benutzergruppe.** *Was ist HL7?* [Online] [Abgerufen am 16. September 2010] http://www.hl7.de/standard/wasist_hl7.php.
- [iHLTemp10].....**Currey, Jane.** *HL7 Templates. HL7 Templates.* [Online] [Abgerufen am 21. September 2010]
<http://www.hl7.org/Library/Committees/template/HL7%20Templates%20-%20Jane%20Currey.ppt>.

- [iKBV06].....**(KBV), Kassenärztliche Bundesvereinigung.** *KBV - IT in der Arztpraxis - Schnittstellen - Weitere Informationen zu xDT.* [Online] 02. Januar 2006. [Abgerufen am 7. Mai 2010] <http://www.kbv.de/ita/4274.html>.
- [iKraka10].....**Krankenkassen Ratgeber.** *Integrierte Versorgung: Innovation durch Vernetzung.* [Online] krankenkassennetz.de GmbH. [Abgerufen am 21. September 2010] <http://www.krankenkassenratgeber.de/integrierte-versorgung/integrierte-versorgung-innovation-durch-verne.html>.
- [iLeitfAr10].....**Verband der Hersteller von IT-Lösungen für das Gesundheitswesen e. V.** *Leitfaden VHitG-Arztbrief.* [Online] [Abgerufen am 06. August 2010] <http://download.vhitg.de/Leitfaden-VHitG-Arztbrief-v150.pdf>.
- [iLOINC10].....**Logical Observation Identifiers Names and Codes (LOINC®).** *Users' Guide.* [Online] [Abgerufen am 19. September 2010] <http://loinc.org/downloads/files/LOINCManual.pdf>.
- [iOID10].....**OID-info.** *Object Identifier (OID) Repository.* [Online] [Abgerufen am 21. September 2010] <http://www.oid-info.com/>.
- [iRIMMod10].....**Stevens, Helen.** *Version 3 From Design to Implementation.* [Online] [Abgerufen am 02. September 2010] http://www.cihi.ca/cihiweb/en/downloads/v3Advanced_AM_e.pdf.
- [iRingh10].....**Ringholm.** *The HL7 roadmap for CDA R3 and the CCD.* [Online] [Abgerufen am 16. September 2010] http://www.ringholm.de/column/hl7_cda_r3_ccd_roadmap.htm.
- [iSciphox01].....**Heitmann, Dr. med. Kai U.** *Deutsches Ärzteblatt: Archiv "SCIPHOX – elektronische Kommunikation im Gesundheitswesen: Auf dem Weg zur integrierten Versorgung".* [Online] 09. November 2001.

- [Abgerufen am 07. Mai 2010]
<http://www.aerzteblatt.de/v4/archiv/artikel.asp?id=29333>.
- [iTon10].....**Schaller, Tony**. *HL7 CDA-CH*. [Online] [Abgerufen am 21. September 2010]
http://www.ehealthsummit.ch/downloads/2008_referate/Schaller_Tony_SEHS2008.pdf.
- [iVHitG10].....**Verband der Hersteller von IT-Lösungen für das Gesundheitswesen**. *VHitG e.V.* [Online] [Abgerufen am 03. September 2010]
http://www.vhitg.de/vhitg/int/02_News_Presse/News.php.
- [iVHitGbrief10]...**Verband der Hersteller von IT-Lösungen im Gesundheitswesen e. V.** *Projekte der Initiative Intersektorale Kommunikation*. [Online] [Abgerufen am 04. September 2010]
http://www.vhitg.de/vhitg/int/06_Ergebnisse_u_Publikationen/Initiative_Intersektorale_Kommunikation/ag_dokumente.php.
- [Johner09].....**Johner, Christian und Haas, Peter**. *Praxishandbuch IT im Gesundheitswesen erfolgreich einführen, entwickeln, anwenden und betreiben*. München : Carl Hanser Verlag, 2009. S. 554.
- [Pet06].....**Haas, Peter**. *Gesundheitstelematik*. Dortmund : Springer-Verlag Berlin Heidelberg, 2006.

Abkürzungsverzeichnis

BLOB	Binary Large Object
CDA	Clinical Document Architecture
CMET	Common Message Element Type
D-MIM	Domain - Message Information Model
Dr.	Doktor
DTD	Document Type Definition
DTL	Data Transformation Language
e.V.	eingetragener Verein
GUI	Graphical User Interface
HDF	HL7 Development Framework
HL7	Health Level 7
HMD	Hierarchical Message Description
ID	Identifikationsnummer
ISO.....	Internationale Organisation für Normung
IT.....	Informationstechnik
ITS	Implementation Technology Specification
LOINC.....	Logical Observation Identifier Names and Codes
MDF	Message Development Framework

MDM Medical Document Management

MIF Model Interchange Format

MSH Message Header

OID Object Identifier

OP Operation

OSI Open Systems Interconnection

PDA Personal Digital Assistant

PDF Portable Document Format

RIM Reference Information Model

R-MIM Refined - Message Information Model

SCIPHOX Standardized Communication of Information Systems
in Physician Offices and Hospitals using XML

UML Unified Modeling Language

VHitG Verband der Hersteller von IT-Lösungen im Gesundheitswesen e.V.

xDT Datentransferprotokolle im ambulanten Bereich

XML eXtensible Markup Language

XSD XML Schema

XSL Extensible Stylesheet Language

XSLT Extensible Stylesheet Language Transformation

Abbildungsverzeichnis

Abbildung I	Prinzip der Intersektoralen Versorgung, eigene Darstellung .4
Abbildung II	Das Ensemble Studio, eigene Darstellung7
Abbildung III	Startseite des Ensemble Management Portals, eigene Darstellung7
Abbildung IV	Aufbau eines HL7 Version 2-MSH-Segments, eigene Darstellung8
Abbildung V	Die 4+2 Basisklassen des RIM, eigene Darstellung..... 11
Abbildung VI	Prinzip der HL7 Schemagenerierung, eigene Darstellung .. 15
Abbildung VII	Interaktionsmodell für das Beispielszenario, eigene Darstellung21
Abbildung VIII	D-MIM für das Beispiel-Szenario, eigene Darstellung 22
Abbildung IX	R-MIM für das Verschicken des Labortestresultates, eigene Darstellung23
Abbildung X	HMD für das Verschicken des Labortestresultates, eigene Darstellung 24
Abbildung XI	XML-Struktur eines CDA-Dokuments, eigene Darstellung ..35
Abbildung XII	Eine Standard HL7-Produktion..... 36
Abbildung XIII	Übersicht über die Produktion zur Transformation einer HL7 Version 2-Nachricht in ein HL7 Version 3-Dokument, eigene Darstellung37

Abbildung XIV	Prinzip der Data Transformation Language mit Nutzung des Visualeditors, eigene Darstellung.....	39
Abbildung XV	Workflow für die Transformation, eigene Darstellung.....	42

Codeverzeichnis

Codeausschnitt I	Aufbau des MSH-Elements einer HL7 Version 2.xml-Nachricht.....	9
Codeausschnitt II	Definieren des Wurzelements und Referenzieren des generierten Schemas durch ein übergeordnetes Schema ..	25
Codeausschnitt III	Die Dokumentenstruktur der Release 1	27
Codeausschnitt IV	Die Dokumentenstruktur von Release 2.....	28
Codeausschnitt V	Aufbau des Sektion-Levels 1.....	29
Codeausschnitt VI	Aufbau des Sektion-Levels 2.....	29
Codeausschnitt VII	Aufbau des Sektion-Levels 3.....	30
Codeausschnitt VIII	Inhalt der HL7 Version 3-Nachrichtenklasse	40
Codeausschnitt IX	Syntax der XML-Dokument Erzeugung.....	41

Tabellenverzeichnis

Tabelle I	Die wichtigsten HL7 Version 3-Datentypen, eigene Darstellung.....	16
Tabelle II	Zu verwendende Datentypen zum Zusammenfassen von Datentypenfolgen, eigene Darstellung	17

Anlagenverzeichnis

Anlage I	Quellcode der Transformationsklasse.....	XVI
-----------------	--	-----

Anlagen

Anlage I Quellcode der Transformationsklasse

```
1| Class Bachelor.Trafos.HL7V2ToV3 Extends Ens.DataTransform [ ClassType = "",  
  ProcedureBlock ]  
2| {  
3| ClassMethod Transform(source As EnsLib.HL7.Message,  
  ByRef target As Bachelor.Message)As %Status  
  {  
4|     set source.DocType="2.5:MDM_T02"  
5|     Set target=##class(Bachelor.Message).%New()  
6|     set writer=##class(%XML.Writer).%New()  
7|     set writer.Charset="UTF-8"  
8|     set writer.Indent=1  
9|     set sc=writer.OutputToStream(target.Content)  
10|    If $$$ISERR(sc) Do $system.OBJ.DisplayError(sc) Quit  
11|    s sc=writer.RootElement("ClinicalDocument")  
12|    //Fest vergeben  
13|    s sc=writer.Element("typeid")  
14|    s sc=writer.WriteAttribute("root", "2.16.840.1.113883.1.3")  
15|    s sc=writer.WriteAttribute("extension", "POCD_HD000040")  
16|    s sc=writer.EndElement()  
17|    //root=$OID($MSH.3,$MSH.4,Document)  
18|    s sc=writer.Element("id")  
19|    s sc=writer.WriteAttribute("root", source.GetValueAt("MSH:21.3"))  
20|    s sc=writer.WriteAttribute("extension", source.GetValueAt("MSH:10"))  
21|    s sc=writer.EndElement()  
22|    //code=$TXA.2 [16]  
23|    s sc=writer.Element("code")  
24|    s sc=writer.WriteAttribute("codeSystem", "2.16.840.1.113883.6.1")  
25|    s sc=writer.WriteAttribute("code", "11488-4")  
26|    s sc=writer.EndElement()  
27|    s sc=writer.Element("title")  
28|    s sc=writer.Write("Klinisches Dokument")  
29|    s sc=writer.EndElement()  
30|    s sc= writer.Element("effectiveTime")  
31|    s sc=writer.WriteAttribute("value", source.GetValueAt("MSH:7"))  
32|    s sc=writer.EndElement()  
33|    s sc=writer.Element("confidentialityCode")  
34|    s sc=writer.WriteAttribute("codeSystem", "2.16.840.1.113883.5.25")  
35|    if (source.GetValueAt("TXA:18") = "U")  
36|    {  
37|        s sc=writer.WriteAttribute("code", "N")  
38|    }  
39|    Else  
40|    {  
41|        s sc=writer.WriteAttribute("code", source.GetValueAt("TXA:18"))  
42|    }  
43|    s sc=writer.EndElement()  
}
```

```
44|         s sc=writer.Element("recordTarget")
45|         s sc=writer.Element("patientRole")
46|         //root=$OID($MSH.3,$MSH.4,Person)
47|         s sc=writer.Element("id")
48|         s sc=writer.WriteAttribute("root", "2.16.840.1.113883.3.933")
49|         s sc=writer.WriteAttribute("extension", source.GetValueAt("PID:3.1"))
50|         s sc=writer.EndElement()
51|         s sc=writer.Element("patient")
52|         s sc=writer.Element("name")
53|         //eventuell
54|         //s sc=writer.WriteAttribute("use", source.GetValueAt("PID:5.7"))
55|         s sc=writer.Element("given")
56|         s sc=writer.Write(source.GetValueAt("PID:5.2"))
57|         s sc=writer.EndElement()
58|         s sc=writer.Element("family")
59|         s sc=writer.Write(source.GetValueAt("PID:5.1"))
60|         s sc=writer.EndElement()
61|         if (source.GetValueAt("PID:5.5") != "")
62|         {
63|             s sc=writer.Element("prefix")
64|             s sc=writer.Write(source.GetValueAt("PID:5.5"))
65|             s sc=writer.EndElement()
66|         }
67|         if (source.GetValueAt("PID:5.4") != "")
68|         {
69|             s sc=writer.Element("suffix")
70|             s sc=writer.Write(source.GetValueAt("PID:5.4"))
71|             s sc=writer.EndElement()
72|         }
73|         //name
74|         s sc=writer.EndElement()
75|         //patient
76|         s sc=writer.EndElement()
77|         //patientRole
78|         s sc=writer.EndElement()
79|         //recordTarget
80|         s sc=writer.EndElement()
81|         s sc=writer.Element("author")
82|         s sc=writer.Element("time")
83|         s sc=writer.WriteAttribute("value", source.GetValueAt("TXA:6"))
84|         s sc=writer.EndElement()
85|         s sc=writer.Element("assignedAuthor")
86|         s sc=writer.Element("id")
87|         //root=$OID($MSH.4,Person)
88|         s sc=writer.WriteAttribute("root", "2.16.840.1.113883.3.933")
89|         s sc=writer.WriteAttribute("extension", source.GetValueAt("TXA:9.1"))
90|         s sc=writer.EndElement()
91|         s sc=writer.Element("assignedPerson")
92|         s sc=writer.Element("name")
93|         s sc=writer.Element("given")
94|         s sc=writer.Write(source.GetValueAt("TXA:9.3"))
95|         s sc=writer.EndElement()
96|         s sc=writer.Element("family")
```

```
97|         s sc=writer.Write(source.GetValueAt("TXA:9.2"))
98|         s sc=writer.EndElement()
99|         if (source.GetValueAt("TXA:9.6") != "")
100|         {
101|             s sc=writer.Element("prefix")
102|             s sc=writer.Write(source.GetValueAt("TXA:9.6"))
103|             s sc=writer.EndElement()
104|         }
105|         if (source.GetValueAt("TXA:9.5") != "")
106|         {
107|             s sc=writer.Element("suffix")
108|             s sc=writer.Write(source.GetValueAt("TXA:9.5"))
109|             s sc=writer.EndElement()
110|         }
111|         //name
112|         s sc=writer.EndElement()
113|         //assignedPerson
114|         s sc=writer.EndElement()
115|         s sc=writer.Element("representedOrganization")
116|         s sc=writer.Element("id")
117|         //root=$OID($MSH.4,Organization)
118|         s sc=writer.WriteAttribute("root", "2.16.840.1.113883.3.933")
119|         s sc=writer.WriteAttribute("extension", source.GetValueAt("TXA:9.14"))
120|         s sc=writer.EndElement()
121|         //representedOrganization
122|         s sc=writer.EndElement()
123|         //assignedAuthor
124|         s sc=writer.EndElement()
125|         //author
126|         s sc=writer.EndElement()
127|         s sc=writer.Element("custodian")
128|         s sc=writer.Element("assignedCustodian")
129|         s sc=writer.Element("representedCustodianOrganization")
130|         s sc=writer.Element("id")
131|         //root=$OID($MSH.4,Organization)
132|         s sc=writer.WriteAttribute("root", "2.16.840.1.113883.3.933")
133|         s sc=writer.WriteAttribute("extension", source.GetValueAt("EVN.7.1"))
134|         s sc=writer.EndElement()
135|         //representedCustodianOrganization
136|         s sc=writer.EndElement()
137|         //assignedCustodian
138|         s sc=writer.EndElement()
139|         //custodian
140|         s sc=writer.EndElement()
141|         s sc=writer.Element("authenticator")
142|         s sc=writer.Element("time")
143|         s sc=writer.WriteAttribute("value", source.GetValueAt("TXA:22.15"))
144|         s sc=writer.EndElement()
145|         s sc=writer.Element("assignedEntity")
146|         s sc=writer.Element("assignedPerson")
147|         s sc=writer.Element("name")
148|         s sc=writer.Element("given")
149|         s sc=writer.Write(source.GetValueAt("TXA:22.3"))
```



```
150|         s sc=writer.EndElement()
151|         s sc=writer.Element("family")
152|         s sc=writer.Write(source.GetValueAt("TXA:22.2"))
153|         s sc=writer.EndElement()
154|         if (source.GetValueAt("TXA:22.6") != "")
155|         {
156|             s sc=writer.Element("prefix")
157|             s sc=writer.Write(source.GetValueAt("TXA:22.6"))
158|             s sc=writer.EndElement()
159|         }
160|         if (source.GetValueAt("TXA:22.5") != "")
161|         {
162|             s sc=writer.Element("suffix")
163|             s sc=writer.Write(source.GetValueAt("TXA:22.5"))
164|             s sc=writer.EndElement()
165|         }
166|         //name
167|         s sc=writer.EndElement()
168|         //assignedPerson
169|         s sc=writer.EndElement()
170|         s sc=writer.Element("representedOrganization")
171|         s sc=writer.Element("id")
172|         //root=$OID($MSH.4,Organization)
173|         s sc=writer.WriteAttribute("root", "2.16.840.1.113883.3.933")
174|         s sc=writer.WriteAttribute("extension", source.GetValueAt("TXA:22.14"))
175|         s sc=writer.EndElement()
176|         //representedOrganization
177|         s sc=writer.EndElement()
178|         //assignedEntity
179|         s sc=writer.EndElement()
180|         //authenticator
181|         s sc=writer.EndElement()
182|         s sc=writer.Element("component")
183|         s sc=writer.Element("structuredBody")
184|         Set i="" For { Set i=source.GetNextIndex("OBXgrp()",i,.pStatus) Quit:i=""
185|             s sc=writer.Element("component")
186|             s sc=writer.Element("section")
187|             s sc=writer.Element("code")
188|             s sc=writer.WriteAttribute("code", source.GetValueAt("OBXgrp(" _i_
189|                 ").OBX:3.1"))
190|             s sc=writer.WriteAttribute("codeSystem", "2.16.840.1.113883.6.1")
191|             s sc=writer.WriteAttribute("codeSystemName", "LOINC")
192|             //code
193|             s sc=writer.EndElement()
194|             s sc=writer.Element("title")
195|             s sc=writer.Write(source.GetValueAt("OBXgrp(" _i_ ").OBX:3.2"))
196|             //title
197|             s sc=writer.EndElement()
198|             s sc=writer.Element("text")
199|             s sc=writer.Write(source.GetValueAt("OBXgrp(" _i_ ").OBX:5"))
200|             //text
201|             s sc=writer.EndElement()
202|             //section
```

```
201|         s sc=writer.EndElement()
202|         //component
203|         s sc=writer.EndElement()
204|     }
205|     //structuredBody
206|     s sc=writer.EndElement()
207|     //component
208|     s sc=writer.EndElement()
209|
210|     If $$$ISERR(sc) Do $system.OBJ.DisplayError(sc) Quit
211|     s sc=writer.EndRootElement()
212|     If $$$ISERR(sc) Do $system.OBJ.DisplayError(sc) Quit
213|     Do target.Content.Rewind()
214|     Quit $$$OK
215| TransformET
216|         Quit $$$ERROR($$$GeneralError, "An error has occurred in
Transform(): " _ $ZError)
217|     }
218| }
```

Selbständigkeitserklärung gem. § 22 Absatz 5 BPO

Hiermit versichere ich, Steffen Hochmuth, dass ich die vorliegende Bachelorarbeit mit dem Titel

Erstellung einer HL7 Version 3-Schnittstelle zur Datenkommunikation im Intersektoralen Bereich am Beispiel eines VHitG-Arztbriefes

selbständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen wurden, sind in jedem Fall unter Angabe der Quelle kenntlich gemacht. Die Arbeit ist noch nicht veröffentlicht oder in anderer Form als Prüfungsleistung vorgelegt worden.

.....
Ort, Datum

.....
Unterschrift

Zusatzseite für Abkürzungen

