

# Inhaltsbasierte Suche in standardisierten elektronischen Gesundheitsakten mittels Archetypen

DISSERTATION

zur Erlangung des akademischen Grades

**Doktor der technischen Wissenschaften**

eingereicht von

**Dipl.-Ing. Christoph Rinner**

Matrikelnummer 0126484

an der  
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Ao.Univ.-Prof. Dipl.-Ing. Dr. Georg Duftschmid

Diese Dissertation haben begutachtet:

---

(Ao.Univ.-Prof. Dipl.-Ing. Dr.  
Georg Duftschmid)

---

(Ao.Univ.-Prof. Mag. Dipl.-Ing.  
Dr. Rudolf Freund)

Wien, 27. August 2013

---

(Dipl.-Ing. Christoph Rinner)



# Erklärung zur Verfassung der Arbeit

Dipl.-Ing. Christoph Rinner  
Schlachthausgasse 30/5/5, 1030 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

---

(Ort, Datum)

---

(Unterschrift Verfasser)



# Danksagung

Ohne die Hilfe und Unterstützung von vielen Helfern wäre das Abschließen des Doktorats nicht möglich gewesen. Alle hier zu nennen ist nicht möglich, jedoch richtet sich mein Dank an euch, für eure Unterstützung und den Glauben an mich.

Bei meinem Betreuer Prof. Georg Duftschmid möchte ich mich herzlich bedanken. Er hat mich während der letzten Jahre am Institut für Medizinisches Informationsmanagement und Bildverarbeitung wissenschaftlich begleitet und hatte für Fragen und interessante Diskussionen immer Zeit.

Weiters möchte ich meinen Kollegen beim Projekt EHR-Arche danken, ohne deren Hilfe große Teile der Doktorarbeit in dieser Form nicht umsetzbar gewesen wären. Speziell möchte ich Michael Kohler danken, mit dem ich mein Büro teilen durfte und der nie müde wurde über meine Arbeit zu diskutieren.

Ein besonderer Dank gilt meiner Familie für die Unterstützung im Verlauf meines gesamten Studiums. Vor allem in der letzten Phase der Doktorarbeit fand sie immer aufmunternde Worte, um mich weiter zu motivieren.



# Abstract

Electronic Health Records (EHRs) should help to handle the increasing amount of data in the health care domain. Shared EHR systems are getting more important to enable an integrated care and the exchange of EHRs between health care providers on a regional and national level. In order to prevent information overload, efficient methods to find relevant information in large EHRs of a patient have to be developed. Current standards in the domain of structured EHRs (i.e. ISO/EN 13606, HL7 CDA, openEHR) apply the so called dual model approach. The EHR content is described by so called *Archetypes* (first model), which specify how the generic classes from the Reference Model (second layer) have to be assembled.

The goal of this work is to show the potential of Archetypes in Shared EHR systems by means of five areas of application. The main focus of the present work is the implementation of a content-based search in the widely-used *Integrating the Health care Enterprise (IHE) Cross Enterprise Document Sharing Profile (IHE XDS)* Shared EHR system architecture. This is an extension to the basic search function from IHE XDS, which only allows retrieval of complete documents by querying a restricted set of document metadata. In contrast the content-based search allows to search medical content within EHR documents and only the relevant parts of EHRs are returned.

In the presented approach all EHR documents are described by Archetypes. Based on these Archetypes possible search terms are offered to the health care professional. The resulting search query is transformed into a standardized meta-data-based search query and a content-based XQuery using the knowledge about the structure of the EHR documents in the Archetypes. The meta-data-based search query is used to retrieve a set of potentially relevant EHR documents from the shared EHR system, the XQuery is used to extract the relevant parts from the EHR documents, which are then visualised. Additionally to the content-based search, an iterative archetype development cycle to support the participation of the health care professionals in the development of archetypes, a method for a *plug-and-play* case form generation to create archetype-based EHR documents as well as a method to semantically validate EHR documents using XML Schema are presented.

Using the iterative archetype development cycle 133 ISO/EN 13606 Archetypes were developed. The *plug-and-play* case form generation was implemented in a web application and 82 archetype-based ISO/EN 13606 EHR extracts were created. They were the basis for an evaluation of the implemented content-based search in an IHE XDS environment.

The evaluation of the implemented content-based search with seven health care professionals has been found fast and stable, it was found more intuitive and faster than conventional meta-data based search in IHE XDS. Archetypes have proven themselves adequate to solve the presented areas of application. The presented content-based search is an add-on to IHE XDS and can be incorporated into existing IHE XDS environments without affecting existing functionality. This work was written in the context of the FWF project *EHR-Arche*.



# Kurzfassung

Elektronische Gesundheitsakten (engl. „Electronic Health Records“, abgekürzt EHRs) sollen helfen das immer größere Datenaufkommen in der Medizin zu bewältigen. Um die integrierte Versorgung und den Austausch von EHR-Daten zwischen Gesundheitsdiensteanbietern (GDAs) auf regionaler und nationaler Ebene zu ermöglichen, gewinnen sogenannte Shared-EHR-Systeme immer mehr an Bedeutung. Es müssen jedoch effiziente Methoden entwickelt werden, um die für eine konkrete Behandlungssituation relevanten Informationen in umfangreichen EHRs von Patienten zu finden und eine Informationsüberflutung der GDAs zu verhindern. Aktuelle Standards zur Spezifikation von strukturierten EHRs (i.e. ISO/EN 13606, HL7 CDA, openEHR) bauen auf dem sogenannten Zwei-Modell-Ansatz auf. Die Inhalte von EHRs werden dabei mittels sogenannter *Archetypen* spezifiziert (erste Modell-Ebene), die aus vordefinierten Komponenten eines Referenzmodells (zweite Modell-Ebene) zusammengesetzt werden.

Ziel der Arbeit ist es anhand von fünf Anwendungsszenarien das Potenzial von Archetypen in Shared-EHR-Systemen aufzuzeigen. Der Hauptfokus der vorliegenden Arbeit liegt auf der Entwicklung einer inhaltsbasierten Suche in einer auf der weit verbreiteten Spezifikation *Integrating the Healthcare Enterprise (IHE) Cross Enterprise Document Sharing (IHE XDS)* basierenden Shared-EHR-Systemarchitektur. Dies stellt eine wesentliche Erweiterung der bisherigen Standard-Suchfunktionalität von IHE XDS dar, welche auf einige wenige Dokument-Metadaten limitiert ist und als Resultat der Suche ausschließlich gesamte EHR-Dokumente zurückliefert. Die inhaltsbasierte Suche ermöglicht es im Gegensatz dazu, nach elementaren Inhalten in EHR-Dokumenten zu suchen und liefert als Ergebnis diese, aus den EHR-Dokumenten extrahierten Inhalte zurück.

Im vorgestellten Ansatz werden sämtliche EHR-Dokumente mittels Archetypen beschrieben und dem GDA darauf basierende Suchbegriffe zur Verfügung gestellt. Die Suchanfrage wird mithilfe der Archetypen in eine standardkonforme metadatenbasierte Suchanfrage, sowie eine an den EHR-Standard angepasste XQuery konvertiert. Durch die metadatenbasierte Suchanfrage werden alle für die Suchanfrage potentiell relevanten EHR-Dokumente geladen, die XQuery extrahiert die Suchergebnisse, die dann für den GDA aufbereitet und visualisiert werden. Ergänzend zur inhaltsbasierten Suche wird ein iterativer Archetypen-Entwicklungszyklus vorgestellt, um GDAs besser in die Entwicklung von Archetypen integrieren zu können, eine Methode zur *Plug-and-Play*-Formulargenerierung, um archetypkonforme EHR-Dokumente erzeugen zu können, sowie eine auf XML-Schema basierende Methode zur semantischen Validierung von EHR-Dokumenten. Mit dem iterativen Archetypen-Entwicklungszyklus wurden 133 ISO/EN 13606 Archetypen erzeugt. Die *Plug-and-Play*-Formulargenerierung wurde in einer Webapplikation umgesetzt, mittels derer 82 archetypbasierte ISO/EN 13606 EHR-Extrakte erzeugt wurden.

Im Zuge der Evaluierung wurde die inhaltsbasierte Suche von sieben GDAs als schnell und ausreichend stabil empfunden, sie sei intuitiver und schneller als die herkömmliche metadatenbasierte Suche in IHE XDS. Archetypen erwiesen sich für die in dieser Arbeit präsentierten Anwendungen als geeignet. Der vorgestellte Ansatz zur inhaltsbasierten Suche stellt eine punktuelle Erweiterung von IHE XDS dar und kann somit in existierende IHE XDS Umgebungen eingebunden werden, ohne bisherige Funktionalitäten zu beeinträchtigen. Die vorliegende Arbeit entstand im Kontext des FWF-Projekts *EHR-Arche*.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Ziel der Arbeit . . . . .	4
1.2	Das <i>EHR-Arche</i> -Projekt . . . . .	6
1.3	Struktur der Arbeit . . . . .	7
<b>2</b>	<b>Grundlagen</b>	<b>9</b>
2.1	EHRs und der Zwei-Modell-Ansatz . . . . .	9
2.1.1	Der Standard ISO/EN 13606 . . . . .	13
2.1.1.1	Das ISO/EN-13606-Referenzmodell . . . . .	15
2.1.1.2	ISO/EN-13606-Archetypen . . . . .	18
2.1.2	HL7 CDA und openEHR . . . . .	22
2.1.3	Archetypen und EHRs . . . . .	25
2.1.3.1	Erzeugen standardisierter, archetypkonformer EHR-Dokumente	25
2.1.3.2	Semantische Validierung von EHR-Dokumenten . . . . .	26
2.1.3.3	Archetyp-Repositorys . . . . .	27
2.2	XML . . . . .	29
2.3	Shared-EHR-System-Architekturen . . . . .	34
2.3.1	Das IHE-Profil Cross Enterprise Document Sharing (XDS) . . . . .	36
2.3.2	Ablauf der Suche nach EHR-Dokumenten in IHE-XDS . . . . .	39
2.3.3	ISO/EN 13606 und IHE-XDS . . . . .	40
2.4	Suchen in EHRs . . . . .	42
2.4.1	Arten von Suchanfragen . . . . .	42
2.4.2	Suche in XML-Dokumenten . . . . .	44
2.4.3	Suche im Semantic Web . . . . .	46
2.4.4	Suche in Shared-EHR-Systemen . . . . .	47
2.4.4.1	Archetypen-unabhängige Suche . . . . .	48
2.4.4.2	Archetypen-spezifische Suche . . . . .	50
2.4.4.3	IHE-Profile für inhaltsbasierte Suchen . . . . .	52
2.4.5	Aufbereitung der Suchergebnisse . . . . .	53
<b>3</b>	<b>Methoden</b>	<b>57</b>
3.1	Inhaltsbasierte Suche in Shared-EHR-Systemen mit zentraler Metadatenkomponente . . . . .	57

3.1.1	Die Wissensbasis . . . . .	60
3.1.2	Die inhaltsbasierte Suchanfrage . . . . .	63
3.1.3	Die metadatenbasierte Suchanfrage . . . . .	64
3.1.4	Die XQuery-Suchanfrage . . . . .	65
3.1.4.1	Auflösen der Suchbegriffe . . . . .	66
3.1.4.2	Bestimmen der Minimal Information Unit . . . . .	66
3.1.4.3	Boolsche Operatoren auflösen . . . . .	67
3.1.4.4	XQuery-Suchanfrage erzeugen . . . . .	68
3.1.5	Aufbereiten der Suchergebnisse . . . . .	68
3.2	Iterativer Archetypen-Entwicklungszyklus . . . . .	69
3.2.1	Initiale Planung . . . . .	70
3.2.2	Erzeugung und Anpassen der Archetypen . . . . .	71
3.2.3	Generierung der Formulare und Dokumente . . . . .	71
3.2.4	Evaluierung und Planung . . . . .	72
3.2.5	Freigabe der Archetypen . . . . .	72
3.3	Plug-and-Play-Formulargenerierung . . . . .	73
3.3.1	Die Model-Komponente . . . . .	74
3.3.2	Die View-Komponente . . . . .	75
3.3.3	Die Controller-Komponente . . . . .	76
3.4	XML-Schema zur semantischen Validierung . . . . .	76
3.4.1	Erfüllung der <i>unique particle attribution constraint rule</i> von XML-Schema . . . . .	78
3.4.2	Erzeugen des XML-Schemas . . . . .	79
3.4.3	Transformation der EHR-Dokumente . . . . .	80
3.4.4	Zusammenfassung der XML-Schema-Validierung . . . . .	80
<b>4</b>	<b>Entwicklung von Archetypen und archetypkonformen EHR-Dokumenten</b>	<b>83</b>
4.1	Die Entwicklung der Archetypen . . . . .	84
4.1.1	Die initiale Planung . . . . .	84
4.1.2	Erzeugen der Archetypen . . . . .	86
4.1.3	Evaluierung und Freigabe der Archetypen . . . . .	89
4.2	Erstellen der archetypkonformen EHR-Dokumente . . . . .	90
4.2.1	Das System ZK-Arche . . . . .	91
4.2.2	Formulargenerierung aus Archetypen . . . . .	93
4.2.3	Erzeugen des <i>comprehensive Archetype</i> . . . . .	94
4.2.4	Die Testdokumente . . . . .	95
4.3	Validierung der Testdokumente . . . . .	96
4.4	IHE-XDS-Testumgebung . . . . .	97
<b>5</b>	<b>Inhaltsbasierte Suche in IHE-XDS</b>	<b>101</b>
5.1	Das ISO/EN-13606- <i>Archetype Repository</i> . . . . .	102
5.1.1	Datenstruktur des <i>Archetype Repository</i> . . . . .	104
5.1.2	Implementierung des <i>Archetype Repository</i> . . . . .	107
5.2	Umsetzung der inhaltsbasierten Suchanfragen . . . . .	108
5.2.1	XML-Struktur der inhaltsbasierten Suchanfrage . . . . .	110

5.2.2	Das Suchinterface . . . . .	114
5.3	Erzeugen der IHE-XDS-Metadatenuche . . . . .	116
5.4	Erzeugen der XQuery-Suchanfrage . . . . .	118
5.4.1	Auflösen der Suchanfrage . . . . .	118
5.4.2	Auflösen der UND- und ODER-Verknüpfungen . . . . .	120
5.4.3	Festlegen der Minimal Information Unit . . . . .	120
5.4.4	Auflösen der Werteinschränkungen . . . . .	121
5.4.5	Generierung der XQuery-Suchanfrage . . . . .	123
5.4.6	Gewünschtes Ergebnis der XQuery-Suchanfrage . . . . .	126
5.4.7	Ausführen der Suche in den EHR-Dokumenten . . . . .	127
5.5	Visualisierung der Suchergebnisse . . . . .	129
5.6	Evaluierung und Laufzeiten . . . . .	131
<b>6</b>	<b>Diskussion</b>	<b>135</b>
6.1	Diskussion der umgesetzten inhaltsbasierten Suche . . . . .	135
6.1.1	Vergleich mit existierenden Suchansätzen . . . . .	139
6.1.2	Alternative Umsetzungen der inhaltsbasierten Suchanfrage . . . . .	140
6.1.3	Optimieren der inhaltsbasierten Suche . . . . .	142
6.1.4	Alternative Visualisierungsansätze . . . . .	146
6.1.5	Nutzung der inhaltsbasierten Suche im Kontext von ELGA . . . . .	148
6.2	Archetypen und das <i>Archetype Repository</i> . . . . .	150
6.3	Die Plug-and-Play Formulargenerierung . . . . .	151
6.4	Die XML-Schema-Validierung . . . . .	153
<b>7</b>	<b>Zusammenfassung und Ausblick</b>	<b>155</b>
	<b>Abbildungsverzeichnis</b>	<b>159</b>
	<b>Tabellenverzeichnis</b>	<b>161</b>
	<b>Verzeichnis der Listings</b>	<b>162</b>
	<b>Literaturverzeichnis</b>	<b>167</b>



# Einleitung

Elektronische Gesundheitsakten, auch als EHRs (vom englischen „Electronic Health Records“) bezeichnet, sollen helfen das immer größere Datenaufkommen in der Medizin zu bewältigen. Neue Diagnosemethoden und die Digitalisierung in der Medizin führen dazu, dass pro Patient immer mehr Daten erhoben werden. Zusätzlich ist die moderne Medizin durch eine steigende Spezialisierung geprägt, die nur umsetzbar ist, wenn die Daten der Patienten zwischen den Gesundheitsdiensteanbietern (GDA) ausgetauscht werden. Aus diesem Grund gewinnen neben den bereits weit verbreiteten lokalen EHR-Systemen [1], in welchen die EHRs der Patienten einer einzelnen medizinischen Einrichtung verwaltet werden, die sogenannten Shared-EHR-Systeme (siehe Abbildung 1.1) zunehmend an Bedeutung. Um eine integrierte Versorgung auf regionaler bis nationaler Ebene zu unterstützen, ermöglichen Shared-EHR-Systeme unterschiedlichen GDAs medizinische Daten aus ihren lokalen EHR-Systemen zu senden und von anderen GDAs zu empfangen.

Im Jahre 2006 wurde von der Bundesgesundheitsagentur eine Machbarkeitsstudie betreffend der Einführung eines nationalen Shared-EHR-Systems in Österreich in Auftrag gegeben [2]. Ziel der Studie war es, die Grundlagen für ein Shared-EHR-System, unter Berücksichtigung der Vorgaben der EU, in Österreich zu schaffen. Die Studie legte die Basis für das momentan in Umsetzung befindliche Shared-EHR-System ELGA [3, 4] sowie für das „ELGA-Gesetz“, das im Oktober 2012 vom Ministerrat beschlossen wurde [5]. Ab 2015 sollen Dokumente zwischen Spitälern und Pflegeanstalten ausgetauscht werden, ab 2016 zusätzlich zwischen Arztpraxen und

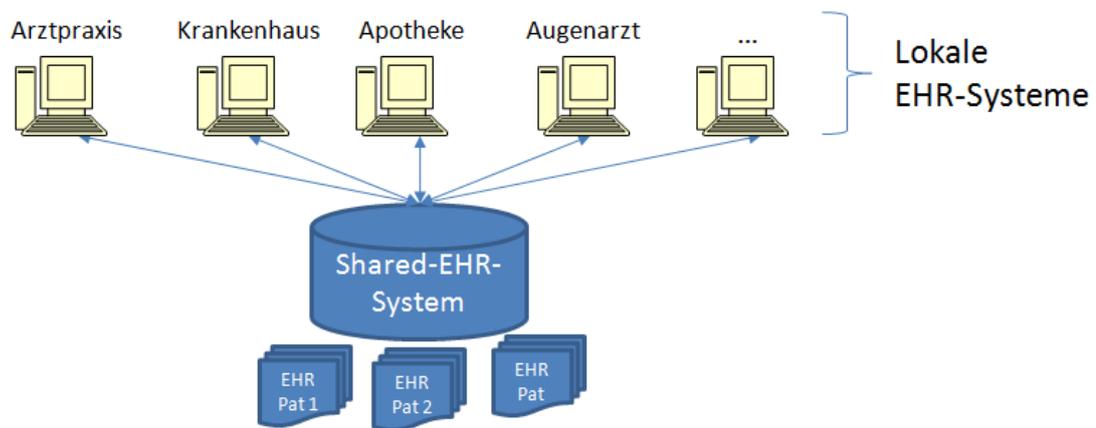


Abbildung 1.1: Überblick über ein Shared-EHR-System. Unterschiedliche GDAs können medizinische Daten eines Patienten mithilfe des Shared-EHR-Systems untereinander austauschen.

Apotheken, ab 2017 folgen die privaten Krankenanstalten.

Obwohl elektronische Systeme zum Verwalten von medizinischen Dokumenten schon seit langem im klinischen Alltag verwendet werden, entstehen bei der Einführung von Shared-EHR-Systemen neue Anforderungen, die berücksichtigt werden müssen. Die Anzahl der Dokumente steigt ebenso wie ihre Heterogenität. Es müssen daher effiziente Methoden entwickelt werden, um Informationen in umfangreichen Patientendokumentationen zu finden [6]. Shared-EHR-Systeme erlauben es, die medizinischen Daten eines Patienten für autorisierte Personen immer und überall zugänglich zu machen. Neben den erwarteten Vorteilen eines Shared-EHR-Systems auf nationaler Ebene befürchten GDAs unter anderem einen erhöhten Zeitaufwand bei der Behandlung des Patienten und die Gefahren von unautorisierten Zugriffen auf die zentral gespeicherten Daten [7]. Die ständige Verfügbarkeit umfangreicher Dokumentenbestände eines Patienten kann weiters zu einer Informationsüberflutung bei den Ärzten führen [8], wodurch die Akzeptanz von GDAs gegenüber Shared-EHR-Systemen weiter vermindert wird. Hochrechnungen für die österreichische ELGA gehen davon aus, dass bei 95% der Patienten weniger als 15 neue EHR-Dokumente pro Jahr hinzukommen und 0,1% mehr als 50 neue EHR-Dokumente pro Jahr haben [9]. Durch geplante zusätzliche Dokumententypen, die in ELGA abgelegt werden, kann davon ausgegangen werden, dass die Zahl der EHR-Dokumente noch steigen wird. Es ist noch nicht gesetzlich geklärt, inwieweit die GDAs dazu verpflichtet werden, alle EHR-Dokumente in Shared-EHR-Systemen eines

Patienten berücksichtigen zu müssen, bzw. wie es im Schadensfall mit der Haftung der GDAs aussieht. Die Haftungsfrage hat einen Einfluss, inwieweit die Informationsüberflutung den Arbeitsalltag der GDAs verändert und Shared-EHR-Systeme von den GDAs akzeptiert werden.

Die Akzeptanz der GDAs von Shared-EHR-Systemen kann durch Filtern und Herauslösen von relevanten Abschnitten aus den EHR-Dokumenten gesteigert werden [10]. In Übereinstimmung damit wird die momentane Implementierung von ELGA vom Präsidenten der Ärztekammer Artur Wechselberger als "Dokumente-Sammelsystem" bezeichnet [11] und die Notwendigkeit effizienter Suchmechanismen in diesem Zusammenhang betont. Im Gesundheitstelematikgesetz §13 Abs. 5 wird die Wichtigkeit der Benutzer- und Anwenderfreundlichkeit sowie der Suchfunktionalität explizit hervorgehoben: "Die ELGA-Systempartner haben unter Berücksichtigung der gebotenen Sicherheitsanforderungen ELGA so zur Verfügung zu stellen, dass die Anbindung von ELGA bei den ELGA-Teilnehmer/inne/n und den ELGA-Gesundheitsdiensteanbietern benutzer- und anwenderfreundlich, insbesondere durch einfach zu handhabende, effektive und für medizinische Kriterien optimierte Such- und Filterfunktionen, möglich ist." [5]

Semantische Interoperabilität wird als ein Grundstein eines EHRs angesehen, um die Qualität und Sicherheit der Patientenbehandlung und der klinischen Forschung sowie das Management von Gesundheitsdiensten zu verbessern [1]. In [12] wird semantische Interoperabilität als die Befähigung kommunizierender EHR-Systeme definiert, ausgetauschte Information auf der Ebene formal definierter Domänenkonzepte zu verstehen. Dieses inhaltliche Verständnis ausgetauschter Information auf Systemebene ermöglicht eine automatische Weiterverarbeitung der Daten, die sowohl dem Arzt als auch dem Patienten einen großen Zusatznutzen (z.B. Entscheidungsunterstützung, einrichtungsübergreifende Datenanalyse, Konsistenzprüfung, etc.) bringen kann. Dieser zusätzliche Nutzen sollte einen gegebenenfalls erhöhten Dokumentationsaufwand (z.B. für eine Vercodung von Information) kompensieren und Ärzte zur Verwendung semantisch interoperabler EHR-Systeme motivieren.

Um die medizinischen Inhalte eines Shared-EHR-Systems weiter analysieren zu können, müssen die EHR-Dokumente semantisch interoperabel dargestellt werden. Zu diesem Zweck existieren unterschiedliche Standards für die Spezifikation strukturierter Gesundheitsdaten (i.e. ISO/EN 13606, HL7 CDA, openEHR), die alle auf dem sogenannten *Zwei-Modell-Ansatz* aufbauen (Details in Kapitel 2.1) und XML als Austauschformat verwenden. Die formale

Definition von Domänenkonzepten wird im *Zwei-Modell-Ansatz* mittels sogenannter *Archetypen* realisiert. Archetypen ermöglichen zu diesem Zweck unter anderem eine strukturelle Beschreibung der EHR-Inhalte, unterstützen die Einbindung von Terminologien, erlauben eine mehrsprachige Datenerfassung und bieten ein Versionierungskonzept.

Eine Literaturanalyse mit 47 Artikeln zeigt, dass Archetypen weltweit immer mehr an Bedeutung gewinnen und als geeignete Möglichkeit angesehen werden, medizinische Daten interoperabel zu speichern [13]. Auch eine Studie des *National Health Service* (NHS) in England, bei der fast 400 Archetypen erzeugt und evaluiert wurden, zeigt, dass Archetypen geeignet sind, medizinisches Wissen darzustellen, und für IT-Projekte der NHS verwendet werden können [14, 15]. Um dem steigenden Datenaufkommen gerecht zu werden und eine optimale Patientenbehandlung zu gewährleisten, werden seit den neunziger Jahren Projekte im Bereich Interoperabilität im Gesundheitsbereich von der Europäischen Kommission und ihren Mitgliedsländern vorangetrieben [1].

## 1.1 Ziel der Arbeit

Ziel der vorliegenden Arbeit ist es, das Potenzial von Archetypen in Shared-EHR-Systemen anhand von fünf Anwendungsszenarien aufzuzeigen. Ein Überblick über die behandelten Anwendungen ist in Abbildung 1.2 zu sehen.

Der Hauptfokus der vorliegenden Arbeit liegt im Umsetzen einer inhaltsbasierten Suche mithilfe von Archetypen in einer Shared-EHR-Systemarchitektur mit zentraler Metadatenkomponente wie etwa *Integrating the Healthcare Enterprise* (IHE) *Cross Enterprise Document Sharing* Profil (IHE-XDS) [16]. IHE-XDS ist international weit verbreitet und wird unter anderem in der geplanten österreichischen ELGA verwendet. IHE-XDS bietet durch die verteilte Speicherung der EHR-Dokumente einen hohen Grad an Datenschutz im Vergleich zu zentraler Speicherung. Die in IHE-XDS verwendete Suche ermöglicht es jedoch nur, ganze EHR-Dokumente mithilfe einiger weniger Metadaten zu suchen. Einfache metadatenbasierte Suchen werden durch das Erstellen eines Index über die Metadaten des Dokuments (wie etwa das Erstellungsdatum, Autor, Dokumenttyp etc.) umgesetzt. Dies hat den Vorteil, dass keine Vorgaben über die Struktur der zu durchsuchenden Dokumente gemacht werden müssen; so können eingescannte Dokumente, PDFs, aber auch strukturierte medizinische Dokumente gesucht werden. Die metadatenbasierte Suche erlaubt es zum Beispiel, alle Laborbefunde in

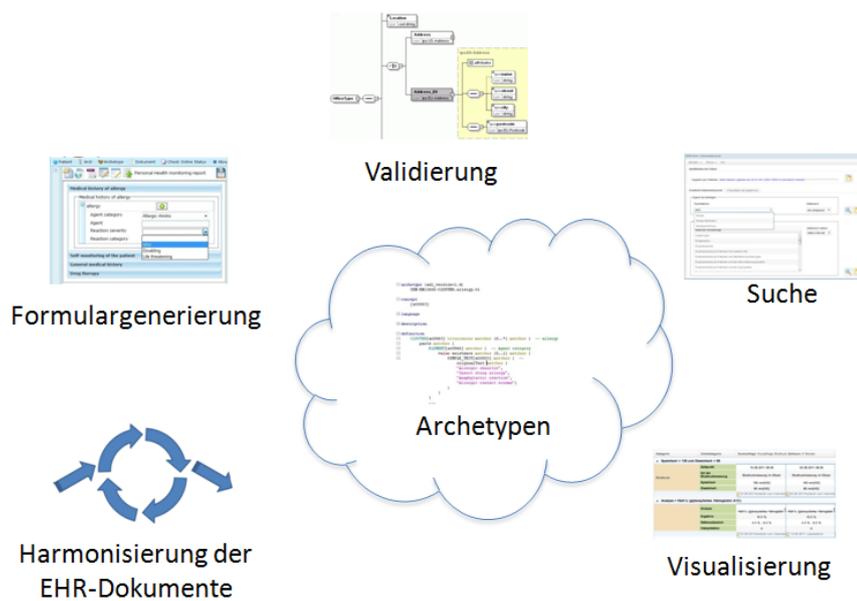


Abbildung 1.2: Archetypen als zentrale Komponente unterschiedlicher Anwendungen in Shared-EHR-Systemen.

einem gewissen Zeitbereich abzurufen. Bei einer inhaltsbasierten Suche ist es zusätzlich möglich direkt Suchbegriffe, die sich auf den medizinischen Inhalt beziehen, anzugeben. So kann in EHR-Dokumenten zum Beispiel nach medizinischen Inhalten wie etwa “alle pathologischen Laborwerte” gesucht werden. Im Gegensatz zur metadatenbasierten Suche, bei der eine Liste der gefundenen Dokumente zurückgeliefert wird, können bei der inhaltsbasierten Suche auch einzelne medizinische Inhalte als Suchergebnis zurückgeliefert werden. Existierende Lösungsansätze einer inhaltsbasierten Suche behelfen sich mit zentralen Indizes der Dokumenteninhalte, wodurch jedoch die verteilte Architektur zum Speichern von EHR-Dokumenten umgangen wird.

Neben der inhaltsbasierten Suche wird ein Ansatz präsentiert, um Formulare automatisch aus Archetypen zu generieren. Dies erlaubt es, vorher nicht bekannte EHR-Dokumente darzustellen und zu editieren. Archetypen werden ebenfalls verwendet, um XML-Schemata zu erzeugen, mithilfe derer erzeugte oder bearbeitete EHR-Dokumente validiert werden können. Die Visualisierung der EHR-Inhalte wird im Rahmen der Ergebnispräsentation der Suche behandelt.

Die in Shared-EHR-Systemen ausgetauschten EHR-Dokumente und deren Inhalte müssen

zwischen den unterschiedlichen GDAs harmonisiert und abgesprochen werden. Archetypen können zur Harmonisierung der EHR-Inhalte verwendet werden. Um GDAs bei der Erzeugung von Archetypen besser einzubinden, wird eine Methode zum Erzeugen von Archetypen präsentiert. Die mit der vorgestellten Methode erzeugten Archetypen werden verwendet, um die automatische Formulargenerierung zu testen und Testdokumente zu erzeugen. Die Testdokumente können mit den aus den Archetypen erzeugten XML-Schemata validiert werden und dienen als Grundlage für das Testen der vorgestellten Suche. Die Suchergebnisse werden mithilfe des Wissens aus den Archetypen aufbereitet und visualisiert.

Ziel der Methode ist es, in existierende EHR-Systeme möglichst wenig einzugreifen und auf existierenden Komponenten wie etwa der Metadaten-Suchanfrage, existierenden XML-Tools und XML-Abfragesprachen etc. aufzubauen.

In meiner Diplomarbeit habe ich mich mit dem Erzeugen von archetypkonformen ISO/EN-13606-EHR-Extrakten aus bestehenden Daten beschäftigt [17, 18, 19]. Der in dieser Arbeit präsentierte Ansatz erweitert die damalige Methodik insofern, als die Daten nicht mehr manuell auf Archetypen gemappt werden müssen, um diese archetypkonform zu exportieren. Die Validierung ist eine Weiterentwicklung, die auf der Diplomarbeit aufbaut, das Erzeugen der Archetypen wird dort nicht angesprochen.

## **1.2 Das *EHR-Arche*-Projekt**

Die vorliegende Arbeit entstand im Kontext des vom *Fonds zur Förderung der wissenschaftlichen Forschung* (FWF) geförderten Projektes *EHR-Arche* (Projektnummer P21396) [20, 21]. Die Ziele des Projektes waren es, 1) die Informationsbedürfnisse von Ärzten im Rahmen einer Diabetes-mellitus-Behandlung zu erheben, 2) die Entwicklung eines Konzepts, um diese Informationsbedürfnisse mit einer herkömmlichen metadatenbasierten Suche in Kombination mit dem Zwei-Modell-Ansatz (siehe Kapitel 2.1) zu erfüllen, 3) das Konzept auf Basis eines Prototyps zu evaluieren.

Der Fokus der vorliegenden Arbeit liegt auf Ziel 2, in dem die Suche basierend auf dem Zwei-Modell-Ansatz umgesetzt wurde. Das Ziel 2 zugeordnete Arbeitspaket befand sich unter Leitung des Autors der vorliegenden Arbeit, der auch maßgeblich an der Umsetzung beteiligt war.

## 1.3 Struktur der Arbeit

In Kapitel 2 werden die in der vorliegenden Arbeit verwendeten Technologien und Standards vorgestellt. Neben einer Einführung in EHRs und den Zwei-Modell-Ansatz werden die Grundlagen von XML beschrieben. Shared-EHR-Systeme und Methoden zur Suche in EHRs werden genauer beleuchtet, da sie den Hauptfokus der Arbeit darstellen. In Kapitel 3 werden die entwickelten Methoden allgemein beschrieben. Kapitel 4 und 5 beschreiben beide die Umsetzungen der zuvor beschriebenen Methoden. In Kapitel 4 werden das Erzeugen der Archetypen, die Erstellung der Testdokumente mithilfe der automatischen Formulargenerierung, das Validieren der Testdokumente sowie die IHE-XDS-Testumgebung beschrieben. In Kapitel 5 wird die Umsetzung der inhaltsbasierten Suche in einer IHE-XDS-Umgebung mit dem neu eingeführten *Archetype Repository* beschrieben. Die Ergebnisse der Evaluierung der Suche im Rahmen des EHR-Arche-Projektes werden ebenfalls hier beschrieben. In Kapitel 6 werden die präsentierten Methoden und Implementierungen diskutiert. In Kapitel 7 wird die Arbeit zusammengefasst und ein Ausblick auf zukünftige Anwendungen der hier vorgestellten Methode gegeben.



# Grundlagen

In diesem Kapitel wird ein Überblick über die verwendeten Technologien in der vorliegenden Arbeit gegeben. Neben dem Standard ISO/EN 13606 (siehe Kapitel 2.1.1) wird auf die allgemeinen Konzepte des Zwei-Modell-Ansatzes eingegangen sowie andere verwandte Standards verglichen. Implementierungen für das Erzeugen, Validieren und Verwalten von EHR-Dokumenten, die auf Archetypen basieren, werden gezeigt. XML und die damit verbundenen Technologien wie XPath, XQuery bilden die Grundlage für den Austausch und die Arbeit mit elektronischen Gesundheitsdaten. Als Basis für den Austausch der Gesundheitsdaten wird die dezentrale Datenhaltungs-Architektur IHE-XDS verwendet. Es wird beschrieben, wie EHR-Dokumente in IHE-XDS abgelegt und geladen werden können.

## 2.1 EHRs und der Zwei-Modell-Ansatz

Durch die zunehmende Spezialisierung in der Medizin werden die engere Zusammenarbeit und der Austausch von Gesundheitsdaten zwischen unterschiedlichen Gesundheitsdiensteanbietern (GDA) (Definition laut [5]) eine wichtige Rolle in einer zukünftigen Gesundheitslandschaft spielen. Die Gesundheitsdaten können dabei in papierbasierter oder elektronischer Form vorliegen. Diese Arbeit konzentriert sich auf elektronische Akten. Elektronische Akten zur Speicherung von medizinischen Informationen können auf unterschiedlichen Standards aufbauen und haben je nach Definition einen anderen Fokus. Ein guter Überblick über die im deutschen Sprachraum verbreiteten Termini in diesem Umfeld (u.a. elektronische Fallakte,

elektronische Krankenakte, elektronische Patientenakte, elektronische Gesundheitsakte) sowie die im Englischen verwendeten Begriffe (u.a. electronic medical record, personal health record, electronic health record, etc.) wird in [22] gegeben.

In weiterer Folge werden die Termini der ISO/TR 20514 [12] verwendet. Die ISO/TR 20514 unterscheidet vier Klassen von *Electronic Health Records* (EHRs) (siehe Abbildung 2.1). Das *Basic-Generic EHR* (BGEHR) bildet den Ausgangspunkt aller EHRs, deren Eigenschaften an die anderen drei Typen vererbt werden. Das BGEHR beschreibt ein EHR, in dem Informationen zu hauptsächlich einem Patienten in computerverarbeitbarer Form abgelegt sind. Es ist explizit auch möglich, Informationen über einen anderen Patienten zu dokumentieren, um z.B. relevante Informationen über die Mutter eines Babys oder genetisch vererbare Krankheiten von Verwandten zu dokumentieren. In der Definition wird bewusst nicht das Wort "elektronisch" verwendet, da Daten auf CD-ROM oder magnetischen Speichermedien nicht wirklich elektronisch abgelegt sind, sondern computerverarbeitbar.

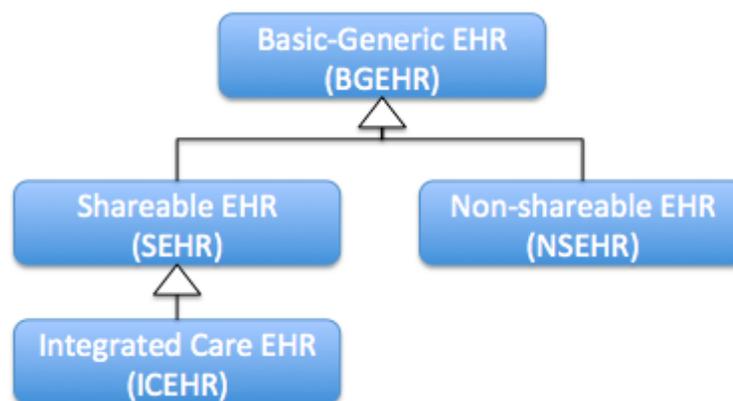


Abbildung 2.1: Die vier Typen von EHRs laut ISO/TR 20514 (aus [12]).

Das *Non-shareable EHR* beschreibt ein BGEHR, das nicht mit anderen Institutionen, Programmen oder Nutzergruppen geteilt wird. Im Gegensatz dazu beschreibt ein *Shareable EHR* ein EHR, das für den Austausch zwischen Institutionen, Programmen und Nutzergruppen konzipiert wurde. Ein *Shareable EHR* sollte auf einem akkordierten logischen Informationsmodell basieren, das nicht nur einem einzelnen Softwareanbieter zugänglich ist. Werden Inhalte eines *Shareable EHRs* zusätzlich zwischen Institutionen ausgetauscht und der primäre Zweck ist eine integrierte medizinische Versorgung, dann wird dieses als *Integrated Care EHR* (ICEHR) bezeichnet. Ein ICEHR sollte auf einem standardisierten logischen

Informationsmodell aufbauen. Da ICEHRs eine Untermenge von Shared-EHRs sind sowie der Begriff Shared-EHR weiter verbreitet ist als ICEHR, wird im Folgenden nur noch von Shared-EHR gesprochen.

Eng verbunden mit der Definition von EHRs ist auch jene von EHR-Systemen, da wie oben erwähnt das Zusammenspiel unterschiedlicher Systeme Auswirkungen auf die Definition des EHRs hat. Ebenfalls in der ISO/TR 20514 [12] werden zwei grundlegende Arten von EHR-Systemen unterschieden, lokale EHR-Systeme, die nur für GDAs einer Institution zugänglich sind, und Shared-EHR-Systeme, die eine integrierte medizinische Versorgung zwischen unterschiedlichen Institutionen unterstützen. Lokale EHR-Systeme beschreiben detaillierte medizinische Inhalte, die primär aus der eigenen Institution stammen, die EHRs können auf allen vier Typen von EHRs aufbauen. In Shared-EHR-Systeme werden im Gegensatz dazu nur ausgesuchte Inhalte bzw. Überblicksdaten aller beteiligten Institutionen abgelegt, weiters muss ein Shared-EHR-System auf dem ICEHR basieren. Im Rahmen des ELGA-Gesetzes [5] wird die elektronische Gesundheitsakte (ELGA) als Informationssystem, das berechtigten Personen Gesundheitsdaten in elektronischer Form orts- und zeitunabhängig zur Verfügung stellt, bezeichnet. Im Sinne von ISO/TR 20514 kann ELGA als Vertreter eines Shared-EHR-Systems gesehen werden. Eine Unterteilung der in dieser Arbeit verwendeten Shared-EHR-System-Architekturen wird in Kapitel 2.3 gegeben.

Eine Schlüsselrolle beim Austausch von Gesundheitsdaten zwischen Shared-EHR-Systemen spielt die Interoperabilität. Interoperabilität ist die Möglichkeit von unterschiedlichen Systemen miteinander zu kommunizieren. Es kann zwischen technischer Interoperabilität, semantischer Interoperabilität und Prozess-Interoperabilität unterschieden werden [23]. Technische Interoperabilität ist dabei die niedrigste Stufe der Interoperabilität und beschäftigt sich mit der Übermittlung von Daten. Die empfangenen Daten können dabei von einer Person verwendet werden, für ein empfangendes Softwaresystem ist ihre inhaltliche Bedeutung aber nicht erkennbar. Die technische Interoperabilität ist eine Voraussetzung für die semantische Interoperabilität. Durch semantische Interoperabilität soll sichergestellt werden, dass die ausgetauschte Information von Sender und Empfänger in genau der gleichen Weise verstanden wird [23]. Um dies zu erreichen, werden medizinischen Klassifikationen und Nomenklaturen wie Snomed CT [24], ein standardisiertes logisches EHR-Informationsmodell sowie ein standardisiertes Modell der EHR-Inhalte verwendet [12]. Die im nächsten Kapitel vorgestellten EHR-Standards decken die Kriterien zum Erreichen von semantischer

Interoperabilität ab. Die Prozess-Interoperabilität baut auf der technischen und der semantischen Interoperabilität auf und koordiniert die Arbeitsabläufe, damit die Geschäftsprozesse in unterschiedlichen Organisationen zusammen funktionieren können [25]. Ziel der Prozess-Interoperabilität ist eine effiziente und sichere Patientenversorgung. Eine Voraussetzung dazu ist die Anpassung der Arbeitsabläufe an die verwendeten EHR-Systeme.

Diese Arbeit beschäftigt sich mit dem semantisch interoperablen Austausch von EHR-Inhalten zwischen Shared-EHR-Systemen.

Die ISO/EN 13606 [26], die HL7 Clinical Document Architecture Release 2.0 (CDA) [27] sowie die openEHR-Spezifikation [28] bilden die drei derzeit wichtigsten Standards für die Spezifikation von strukturierten EHR-Inhalten. Alle drei Standards bauen auf einem logischen Informationsmodell auf und können zur Beschreibung der Gesundheitsdaten in Shared-EHRs verwendet werden. Im Folgenden werden die drei Standards beschrieben. Diese Arbeit legt den Hauptfokus auf die ISO/EN 13606, weshalb dieser Standard am ausführlichsten behandelt wird.

Bei der Modellierung von Gesundheitsdaten werden zwei grundlegende Ansätze unterschieden. Der Ein-Modell-Ansatz (AKA Gesamtmodellansatz), der Information und Wissen in einem gemeinsamen Modell vereint, sowie der Zwei-Modell-Ansatz, der Information und Wissen in unterschiedliche Modelle aufteilt, um EHR-Inhalte zu modellieren (siehe Abbildung 2.2) [29]. Der Ein-Modell-Ansatz eignet sich gut zur Modellierung einfacher, klar abgrenzbarer Anwendungsbereiche, durch das ständig wachsende Wissen im medizinischen Bereich und die sich dadurch verändernden Anforderungen ist die Wartung und Anpassung eines großen Gesamtmodells jedoch unpraktikabel. Die drei Standards ISO/EN 13606, CDA sowie openEHR bauen daher auf dem Zwei-Modell-Ansatz auf.

Im Zwei-Modell-Ansatz wird die Information mittels eines logischen Informationsmodells modelliert. Somit fallen EHR-Dokumente, die mit dem Zwei-Modell-Ansatz modelliert sind, unter die Definition eines Shared-EHRs. Das Informationsmodell besteht aus einigen wenigen generischen Klassen, mit deren Hilfe beliebige medizinische Inhalte dargestellt werden können. Das Informationsmodell der CDA ist das *Refined-Message Information Model* (R-MIM), im Standard ISO/EN 13606 und der openEHR-Spezifikation das *Referenzmodell*. Mithilfe des zweiten Modells, welches das Wissen abbildet, wird spezifiziert, wie die generischen Klassen des Informationsmodells verwendet werden müssen, um konkrete medizinische Inhalte darzustellen. Diese Wissensmodelle können als Freitext wie in den

CDA-Implementierungsleitfäden (z.B. die Implementierungsleitfäden von ELGA [4]) sowie als computerverarbeitbare Strukturen wie Archetypen (ISO/EN 13606 und openEHR) oder HL7 Templates [30] dargestellt werden. Abhängig von der Modellierung kann beim Zwei-Modell-Ansatz mehr oder weniger Wissen im Informationsmodell abgebildet werden [31]. Die CDA hat im Vergleich zur ISO/EN 13606 ein größeres Informationsmodell, so werden Konzepte wie Medikation etwa schon direkt im Informationsmodell spezifiziert. Dies erleichtert das Modellieren der EHR-Inhalte, da gewisse Aspekte schon vom Informationsmodell abgedeckt sind, die Konzepte können jedoch nicht mehr einfach geändert bzw. ergänzt werden.

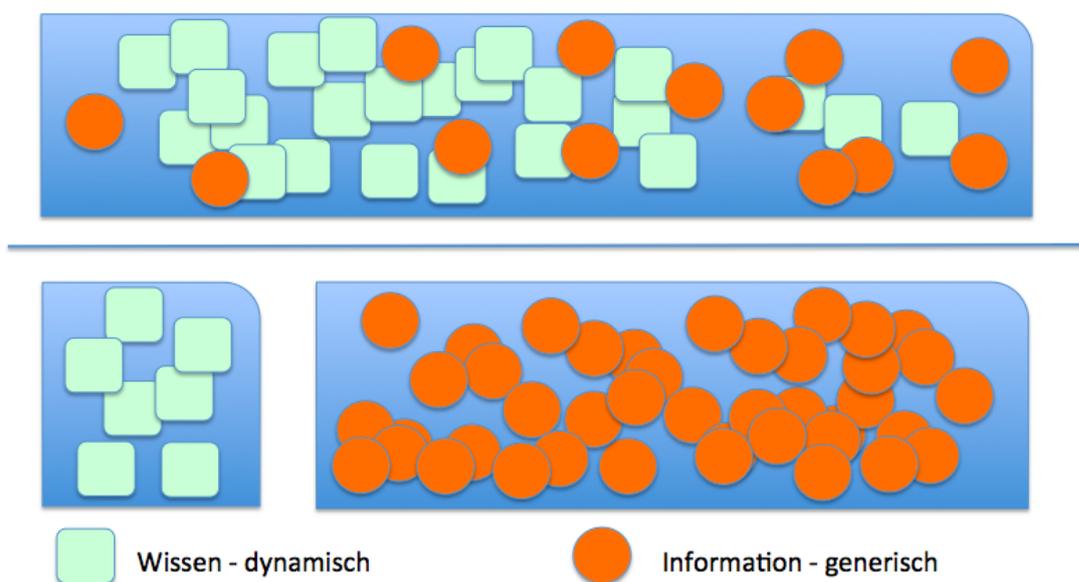


Abbildung 2.2: Wissen und Information im Ein-Modell-Ansatz (oben) und Zwei-Modell-Ansatz (unten).

### 2.1.1 Der Standard ISO/EN 13606

Die ISO/EN 13606 ist ein recht junger Standard. 2010 wurde der fünfte und letzte Teil des Standards ISO/EN 13606 als offizieller Standard aufgenommen. Durch die weite Verbreitung der CDA, die in direkter Konkurrenz zur ISO/EN 13606 steht, sowie durch die zu großen Teilen idente Spezifikation von openEHR existieren zur Zeit nur wenige Implementierungen des Standards ISO/EN 13606. In einer Machbarkeitsstudie aus dem Jahr 2007 betreffend der Einführung von ELGA [2] wurde die zu dieser Zeit nur als "Vorstandard" vorliegende prEN

13606 mit der CDA verglichen. Für eine Umsetzung in ELGA wurde eine Empfehlung für die CDA ausgesprochen.

Mit der Gründung der *EN 13606 Association* [32] wurde eine Plattform geschaffen, um Information bezüglich des Standards ISO/EN 13606 für GDAs sowie Hersteller von Gesundheitssoftware an einem zentralen Ort zur Verfügung zu stellen. Die gemeinnützige Organisation soll zusätzlich die Implementierung und Weiterentwicklungen des Standards ISO/EN 13606 koordinieren und vorantreiben. Auf der Homepage der Organisation werden einige Implementierungen des Standards ISO/EN13606 vorgestellt, die hier kurz zusammengefasst werden. Im brasilianischen Bundesstaat Minas Gerais ist geplant, ISO/EN-13606-EHR-Extrakte und ISO/EN-13606-Archetypen zur Spezifikation von EHR-Inhalten zu verwenden [33]. Auf der Homepage des Projektes [34] sind auch ISO/EN-13606-Archetypen (u.a. für Allergie, Ernährung, Medikamente) veröffentlicht. Die Daten werden dabei nicht mit IHE-XDS zur Verfügung gestellt, sondern auf einem zentralen Server gespeichert. Im *ByMedConnect*-System [35, 36] werden ebenfalls ISO/EN-13606-EHR-Extrakte verwendet, um Daten zwischen Arztpraxen in Bayern auszutauschen. Im Rahmen des *EPSOS*-Projekts [37] der Europäischen Kommission wurde eine technische Spezifikation zur ISO/EN 13606 veröffentlicht [38]. Die für die CDA modellierten Dokumenttypen, i.e. elektronisches Rezept (engl. *ePrescription*), die elektronisch unterstützte Medikamentenausgabe (engl. *eDispensation*) sowie das Patientendossier (engl. *Patient Summary*), werden darin mithilfe von ISO/EN-13606-Archetypen beschrieben.

Nach den im Folgenden beschriebenen ersten beiden Teilen des Standards ISO/EN 13606, die sich mit dem Referenzmodell und den Archetypen beschäftigen, werden in Teil 3 des Standards *Reference archetypes and term lists* Codelisten, die im Referenzmodell verwendet werden, angegeben. Zusätzlich sind Regeln zum Konvertieren von openEHR und CDA in ISO/EN 13606 angegeben. Im vierten Teil *Interface specification* und im fünften Teil *Security* werden die Nachrichten zum Austausch sowie die Regelung von Zugriffsrechten auf ISO/EN-13606-EHR-Extrakte geregelt. Es werden nur die ersten drei Teile beschrieben, da Teil 4 und 5 durch die Verwendung der IHE-XDS-Architektur abgedeckt werden.

### 2.1.1.1 Das ISO/EN-13606-Referenzmodell

Das zum Erreichen semantischer Interoperabilität benötigte standardisierte logische Informationsmodell wird im ersten Teil des Standards ISO/EN 13606 beschrieben und als Referenzmodell bezeichnet. Das Referenzmodell wird in das *Extrakt*-Paket (siehe Abbildung 2.3) und das *Demografie*-Paket unterteilt. Zusätzlich werden noch diejenigen Datentypen aus dem Standard CEN TS 14796:2003 [39], die im Referenzmodell direkt vorkommen, beschrieben.

Die EHR\_EXTRACT-Klasse ist die Ausgangsklasse der EHR-Extrakte. Sie dient dazu, Informationen über das EHR-Extrakt und das Erstellungsdatum, unterschiedliche IDs für das EHR-Extrakt selber, den Patienten und das System, in dem das EHR-Extrakt erzeugt wurde, anzugeben. Diese Information kann mit dem *Header* in CDA-Dokumenten verglichen werden (siehe Kapitel 2.1.2).

Instanzen des ISO/EN-13606-Referenzmodells werden als EHR-Extrakt bezeichnet. Ein EHR-Extrakt kann dabei das gesamte EHR eines Patienten abbilden oder nur einen Teilaspekt darstellen. Werden Teile eines EHRs extrahiert und als eigene EHR-Extrakte weitergegeben, so wird über die EXTRACT\_CRITERIA-Klasse beschrieben, welche Teile wie entnommen wurden. Es kann unter anderem der Zeitbereich des EHR-Extraktes dokumentiert werden oder ob rekursiv alle Versionen und Änderungen mit extrahiert werden. Im *Extrakt*-Paket werden nur IDs zu Personen angegeben. Über die Klassen im Demografie-Paket werden die Rollen (z.B. Patient, Arzt), Namen, Adressen, zugehörige Organisation oder weitere IDs wie zum Beispiel die Sozialversicherungsnummer einer Person festgelegt und mit der im *Extrakt*-Paket verwendeten ID verknüpft. Eine Person kann so im EHR-Extrakt an mehreren Stellen eindeutig referenziert werden.

Die Klassen, die medizinische Inhalte abbilden, bauen auf den Vorgaben der ISO TS 18308 [40] auf, die Anforderungen an eine EHR-Architektur beschreibt. EHR-Dokumente werden dabei so wie ihre papierbasierten Pendanten in Ordner, Dokumente, Überschriften sowie die medizinischen Inhalte strukturiert. EHR-Extrakte beinhalten Instanzen der COMPOSITION-Klasse, die wahlweise mithilfe der FOLDER-Klassen weiter gruppiert werden können. Eine COMPOSITION wird für medizinische Informationen verwendet, die von einem einzelnen GDA stammen und eine Dokumentation eines einzelnen Arztbesuches darstellen oder im Rahmen einer Dokumentations-Session gemacht werden. Die

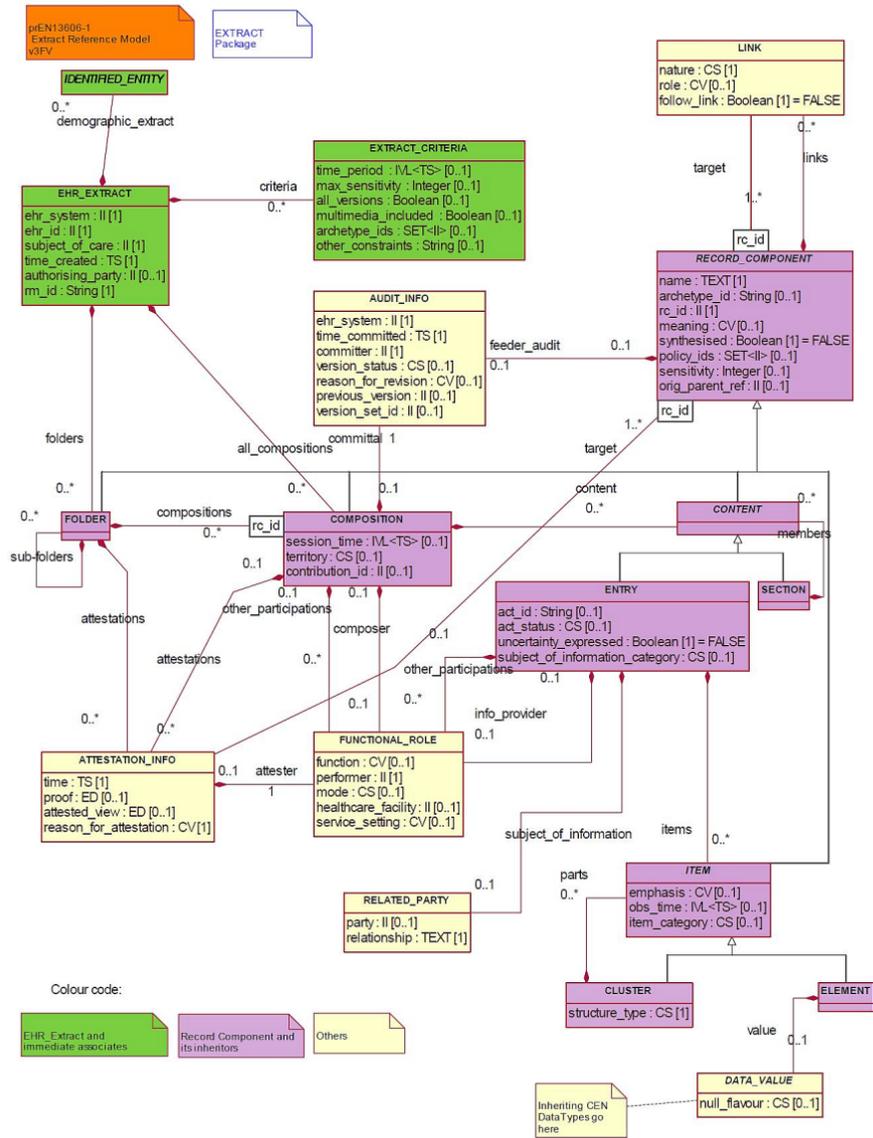


Abbildung 2.3: UML-Modell des ISO/EN-13606-Extrakt-Pakets (aus [26]).

COMPOSITION-Klasse kann in etwa mit den Dokumenttypen, die in der ELGA verwendet werden (i.e. Arztbrief, Laborbefund etc.), verglichen werden und entspricht somit im Umfang dem eines CDA-Dokumentes, wobei einige Metadaten schon in der EHR\_EXTRACT-Klasse abgebildet werden. Eine COMPOSITION kann Informationen bezüglich der Speicherung (i.e. Datum, Autoren, etc.) und der Versionierung enthalten. Auf der Ebene der COMPOSITION-Klasse bzw. FOLDER-Klasse ist es möglich, Inhalte eines EHR-Extrakts direkt zu attestieren, weitere Klassen können über deren ID attestiert werden.

Die COMPOSITION-Klasse kann SECTION- oder ENTRY-Klassen enthalten. SECTIONs können weitere SECTIONs oder ENTRYs enthalten und dienen als Unterteilungen bzw. Überschriften. ENTRYs beschreiben den eigentlichen medizinischen Inhalt. Über das *subject\_of\_information\_category*-Attribut und die *subject\_of\_information*-Beziehung der ENTRY-Klasse kann sich ein ENTRY und alle darunter befindlichen Klassen auf unterschiedliche Personen oder Objekte beziehen (z.B. Familienanamnese). ENTRYs werden für Beobachtungen, Schlussfolgerungen, Feststellungen und Aktionen verwendet. Über CLUSTER und das *structure\_type*-Attribut können Tabellen und Listen in ENTRYs eingebunden werden. Über die ELEMENT-Klasse werden Einzelwerte abgebildet. Es können dafür alle in [39] definierten Datentypen verwendet werden.

Die Klassen FOLDER, COMPOSITION, SECTION, ENTRY, CLUSTER und ELEMENT erben die Eigenschaften der abstrakten Klasse *RECORD\_COMPONENT*. Die *RECORD\_COMPONENT* erweitert die Klassen um Metainformationen wie IDs, Versionsinformationen oder Links, um Beziehungen zwischen medizinischen Inhalten herzustellen. An dieser Stelle wird auch die Verbindung zwischen Referenzmodell und Archetypen über das Attribut *archetype\_id* hergestellt. Jeder *RECORD\_COMPONENT* kann so ein eindeutiger Archetyp-Knoten zugewiesen werden.

In Shared-EHR-Systemen werden einzelne EHR-Dokumente abgelegt, dies entspricht einem EHR-Extrakt, welches auf der COMPOSITION-Ebene beginnt. Die Bezeichnung wird in der vorliegenden Arbeit als Synonym für ein COMPOSITION-EHR-Extrakt bzw. CDA-Dokument verwendet.

Im Standard ISO/EN 13606 werden keine Angaben, wie EHR-Extrakte persistiert werden sollen, gemacht. In der vorliegenden Arbeit werden EHR-Extrakte durchgehend als XML-Dokumente gespeichert. Das XML-Schema der *EN 13606 Association* [32] wird verwendet.

### 2.1.1.2 ISO/EN-13606-Archetypen

Im Standard ISO/EN 13606 werden Information, im generischen Referenzmodell, und medizinisches Wissen in Archetypen aufgeteilt. Archetypen können dabei elementare medizinische Konzepte wie z.B. Blutdruck, aber auch komplexe Konzepte wie die Inhalte und Strukturen von ganzen Dokumenttypen (z.B. Arztbrief) beschreiben. Archetypen beschreiben dabei, wie die Klassen des Referenzmodells zusammengesetzt werden dürfen bzw. müssen. Dieser Zusammenhang von Referenzmodell und Archetypen lässt sich sehr gut mit einer Lego-Metapher beschreiben. Die unterschiedlichen Klassen des Referenzmodells können mit unterschiedlichen Arten von Legosteinen verglichen werden, die Archetypen mit den Lego-Bauplänen. Die Baupläne geben an, wie die Legosteine angeordnet werden müssen, um gewisse Objekte wie Häuser, Autos etc. zu bauen. Der Bauplan stellt sicher, dass die Objekte nach einem vorgegebenen Muster erzeugt werden. Dieses Muster, das mittels Archetypen vorgegeben wird, ist wichtig, damit medizinische Inhalte beim Empfänger richtig interpretiert und verstanden werden können.

Archetypen stellen im Gegensatz zu den als Freitext zur Verfügung gestellten CDA-Implementierungsleitfäden das Wissen über den Aufbau und die Struktur von EHR-Inhalten in computerverarbeitbarer Form zur Verfügung. Alle Archetypen sind Instanzen des im 2. Teil des Standards ISO/EN 13606 vorgestellten *Archetype Object Model* (AOM). Instanzen des AOM werden im Standard ISO/EN 13606 in textueller Form in der formalen Sprache *Archetype Definition Language* 1.4 (ADL) persistiert. Ein Archetyp in ADL wird in die verpflichtenden Sektionen *concept*, *language*, *definition* und *ontology* sowie in die optionalen Sektionen *specialize*, *description*, *invariant* und *revision\_history* unterteilt (siehe Listing 2.1). *Invariant* und *revision\_history* wurden im Rahmen der vorliegenden Arbeit nicht verwendet und sind im Beispiel nicht angegeben. Die *Invariant*-Sektion dient dazu, Aussagen über Ausdrücke in der *definition*-Sektion zu machen, die dort nicht möglich sind, wie etwa Formeln, oder Ausdrücke, die mehrere Knoten betreffen. In der *revision\_history*-Sektion werden die Änderungen in unterschiedlichen Versionen des Archetyps dokumentiert.

Ist ein Archetyp eine Spezialisierung eines anderen Archetypen, wird dies in der *specialize*-Sektion angegeben. Eine Spezialisierung ist dabei nur von höchstens einem Archetypen möglich. Spezialisierungen können verwendet werden, um Archetypen noch zusätzlich einzuschränken. Der in Listing 2.1 dargestellte Archetyp für "Glukosestatus"

Listing 2.1: Beispiel ADL des Labortests für Glukosestatus.

```

1 archetype (adl_version=1.4)
2   CEN-EN13606-ENTRY.lab_test_Glucose_status.v1
3 specialize
4   CEN-EN13606-ENTRY.lab_test.v1
5 concept
6   [at0000.1]
7 language
8   original_language = <[ISO_639-1::de]>
9   ...
10 description
11   ...
12 definition
13 ENTRY[at0000.1] matches { -- Glukosestatus/Glucose status
14   items matches {
15     CLUSTER[at0001] matches { -- CLUSTER
16       parts matches {
17         ELEMENT[at0008.1] matches { -- Analyse/Analysis
18           value matches {
19             SIMPLE_TEXT[at0064] matches {
20               originalText existence matches {0..1} matches {[local::
21                 at0840, at0844, ...]}]}
22             allow_archetype ELEMENT[at0003] occurrences matches {0..1} matches
23               {
24                 include archetype_id/value matches {/CEN-EN13606-ELEMENT.
25                   lab_test_result.v1/}}
26             allow_archetype ELEMENT[at0007] occurrences matches {0..1} matches
27               {
28                 include archetype_id/value matches {/CEN-EN13606-ELEMENT.
29                   lab_test_reference_range.v1/}}
30             allow_archetype ELEMENT[at0005] occurrences matches {0..1} matches
31               {
32                 include archetype_id/value matches {/CEN-EN13606-ELEMENT.
33                   lab_test_interpretation.v1/}}
34 ontology
35   term_definitions = <
36     ["de"] = <
37       items = <
38         ["at0840"] = <text = <"Blut-Glukosewert">>
39         ["at0844"] = <text = <"HbA1c (glykosyliertes Haemoglobin A1C)">>
40         ["at0008.1"] = <text = <"Analyse">> ...
41   term_binding = <
42     ["AAArche"] = <
43       items = <
44         ["at0844"] = <[AAArche::86.6]>
45         ["at0008.1"] = <[AAArche::86.1]> ...

```

spezialisiert den “Labor”-Archetyp. Im “Labor”-Archetyp können unterschiedliche Analysen angegeben werden, im “Glukosestatus”-Archetyp können nur noch die Analysen des “Glukosestatus” ausgewählt werden. Die Sektionen *concept*, *language* und *description* dienen dazu, den Inhalt eines Archetypen zu beschreiben. Diese Metadaten beinhalten unter anderem die Angabe der unterschiedlichen Sprachen, in denen der Archetyp verfasst und übersetzt ist, Informationen über die Autoren oder ob es sich um eine fertige Version oder eine “Vorversion” des Archetyps handelt.

In der *definition*-Sektion wird das eigentliche medizinische Wissen als auf dem Referenzmodell basierende Baumstruktur abgebildet. Es werden dabei nur diejenigen Teile im Archetyp erwähnt, die das Referenzmodell weiter einschränken. Der Wurzelknoten der *definition*-Sektion bezieht sich immer auf eine Klasse des Referenzmodells, gefolgt von einem Knoten, der die Attribute des Referenzmodells einschränkt. Klassen und Attribute wechseln sich immer ab. Knoten, die Klassen einschränken, haben immer eine Archetyp-Knoten-ID der Form “AT” + Ziffern mit optionalen Punkten zwischen den Ziffern, die in eckigen Klammern angegeben wird (z.B. [at0000.1]). Über die Archetyp-Knoten-ID und die Attribute können eindeutige Pfade, sogenannte Archetyp-Pfade, aufgebaut werden, mit denen jeder Knoten eindeutig identifiziert werden kann. Über *Slots* können andere Archetypen eingebunden werden. Es können dabei einzelne Archetypen direkt angegeben oder über reguläre Ausdrücke festgelegt werden, welche Archetypen eingebunden werden dürfen. Teile innerhalb eines Archetyp können mittels des Keywords *use\_node* und dem Archetyp-Pfad des jeweiligen Archetyp-Knoten mehrfach innerhalb eines Archetyp wiederverwendet werden. Die Keywords *cardinality*, *occurrences* und *existence* werden verwendet, um die Anzahl der erlaubten nachfolgenden Elemente anzugeben bzw. für Klassen und Attribute festzulegen, wie oft sie instanziiert werden dürfen.

In der *ontology*-Sektion werden der jeweiligen Archetyp-Knoten-ID Namen und Beschreibungen - unter Umständen in verschiedenen Übersetzungen - hinzugefügt. Ein Archetyp kann aus einer großen Anzahl von frei definierten Archetyp-Knoten zusammengesetzt werden. Um diesen Knoten eine semantische Bedeutung zuweisen zu können, also die Verbindung mit einer medizinischen Klassifikation bzw. Terminologie herstellen zu können, wird der *term\_binding*-Abschnitt innerhalb des *ontology*-Abschnitt bereitgestellt. Jeder Archetyp-Knoten-ID können darin beliebig viele Codes zugewiesen werden. Ein Beispiel ist in Listing 2.1 und der Archetyp-Node-ID *at0844* zu sehen. Der Archetyp-Node-ID ist der Text *HbA1c* zugewiesen sowie der Code *AAA arche:86.6*, der auf den

Code 86.6 der für das EHR-Arche-Projekt entwickelte Terminologie *AAArche* verweist. In dieser Terminologie wird jedem erhobenen Informationsbedürfnis ein Code zugewiesen.

Der *term\_binding*-Mechanismus wird in einigen wissenschaftlichen Arbeiten behandelt. Das in [41] beschriebene Tool *Model Standardisation using Terminology* (MoST) dient dazu, relevante SNOMED-CT-Codes zu medizinischen Konzepten in Archetypen zu finden. Basierend auf den im Archetyp spezifizierten medizinischen Konzepten werden von *MoST* relevante SNOMED-CT-Codes gesucht, diese können dann mittels *term\_binding* einem Archetypknoten zugeordnet werden. In [42] wird ein Archetyp-Editor präsentiert, der speziell auf das Mapping von existierenden medizinischen Terminologien abgestimmt wurde. Dazu wird das *MoST*-Tool mit einem zusätzlichen Tool, *TermViz* [43] zum Visualisieren von medizinischen Terminologien wie SNOMED-CT oder ICD10, im Archetyp-Editor kombiniert. In [44] wird der Begriff des *terminological Shadow* eingeführt, dabei werden zu allen Archetypknoten passende SNOMED-CT-Codes gesucht. Der Ansatz basiert auf der Annahme, dass in SNOMED-CT rund 80% des medizinischen Wissens abgebildet sind und die zweite Hierarchieebene mit 345 Kategorien eine ausreichende Granularität besitzt, um Aussagen über die abgedeckten medizinischen Fachgebiete zuzulassen. Mithilfe dieser Methode kann ein Überblick gewonnen werden, welche Fachgebiete in allen Archetypen eines Archetyp-Repository abgebildet werden. Dazu können dann relevante SNOMED-CT-Codes gefunden werden. Ebenfalls für die Analyse von Archetypen wird in [45] eine Methode basierend auf der *Unified Medical Language System UMLS* [46] vorgestellt, um Ähnlichkeiten zwischen existierenden Archetypen herauszufinden.

Der Standard ISO/EN 13606 beinhaltet keine Angaben zum weiteren Einschränken von Archetypen für einen spezifischen Anwendungsfall, wie dies bei openEHR mithilfe der openEHR-Templates möglich ist. Um die für einen Anwendungsfall nicht benötigten Teile eines Archetypen auszublenden, können in openEHR-Templates die *existence* sowie die *occurrence* verändert werden, *Slots* können zusätzlich eingeschränkt werden sowie mögliche Ausprägungen in Archetypen und *term\_bindings* noch weiter eingeschränkt werden. Dies ist auch innerhalb eines ISO/EN-13606-Archetyps mit einer Spezialisierung möglich, in openEHR-Templates kann dies jedoch über mehrere Archetypebenen definiert werden. OpenEHR bietet diverse Tools an, um mit openEHR-Templates zu arbeiten. Es gibt auch in der *EN 13606 Association* Bestrebungen, Einschränkungen über mehrere Archetypebenen zu definieren und alle COMPOSITION- bzw. SECTION-Archetypen automatisch als Templates

zu interpretieren, dies würde aber erst in einer zukünftigen Version der ISO/EN 13606 berücksichtigt werden.

Wie im weiteren Verlauf beschrieben bildet das in computerverarbeitbarer Form beschriebene Wissen in Archetypen die Grundlage für verschiedene EHR-spezifische Anwendungen.

### **2.1.2 HL7 CDA und openEHR**

Die CDA ist für den Austausch von klinischen Dokumenten konzipiert und ist als ANSI- und ISO-Standard veröffentlicht. CDA-Dokumente werden in Krankenhäusern wie etwa im AKH-Wien für den intramuralen Datenaustausch [47], auf nationaler Ebene wie zum Beispiel in der österreichischen Gesundheitsakte ELGA sowie auf internationaler Ebene wie etwa im *EPSOS*-Projekt [37], welches von der Europäischen Kommission kofinanziert wird, verwendet. Die CDA wird in vielen Shared-EHR-System-Architekturen wie IHE-XDS als Dokumentenformat verwendet. CDA-Dokumente sind immer XML-Dokumente, die mit einem XSLT-Stylesheet in eine von Menschen lesbare Form gebracht werden können. Die CDA-Dokumente sind in einen *Header* und einen *Body* aufgeteilt. Der *Header* enthält Metainformationen, die unter anderem den Typ und den Inhalt des Dokumentes beschreiben, aber auch Informationen zum Patienten, Autor, Empfänger, etc. enthalten. Die medizinischen Inhalte werden im *Body* abgebildet.

Im *CDA-Body* können vier unterschiedliche Strukturierungsgrade unterschieden werden. Level 0 beinhaltet die medizinischen Inhalte in unstrukturierter Form, die als Binärdatei eingebunden werden. Level 1 wird als teilstrukturiert bezeichnet. Die Struktur der medizinischen Inhalte wird dabei direkt in XML abgebildet. Die medizinischen Inhalte werden jedoch weiter in unstrukturierter Form als Text in XML-Elemente eingebettet. In Level 2 werden die XML-Elemente in Level 1 zusätzlich mit standardisierten Codesystemen klassifiziert. Level 3 bezeichnet vollstrukturierte CDA-Dokumente, die zusätzlich zu Level 2 die medizinischen Inhalte mithilfe der Klasse *entry* und den Klassen des *clinicalStatement* vollstrukturiert abbilden. Werden die Inhalte vollstrukturiert dokumentiert, wird für die XSLT-Transformation der Visualisierung auf die unstrukturierten Texte von Level 2 zurückgegriffen. In Listing 2.2 wird der Level-2-Text und derselbe Inhalt als vollstrukturierte Instanzen des *clinicalStatement* dargestellt. Der Ausschnitt stammt aus einem Level-2-CDA-Dokument und beschreibt das Material (Zeile 12 und Zeile 24) in einem Laborbefund. Der unstrukturierte Wert wird in Zeile 15, der strukturierte Wert in den Zeilen 25–28 angeführt.

Listing 2.2: Beispiel CDA Level 2 und Level 3 (gekürzt aus [4])

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <ClinicalDocument xmlns="urn:hl7-org:v3">
3   ...
4   <component typeCode="COMP">
5     <structuredBody classCode="DOCBODY">
6       <component typeCode="COMP">
7         <section classCode="DOCSECT">
8           <title>Probeninformation</title>
9           <text>
10            <table>
11              <thead>
12                <tr><th>Spezimen</th><th>Materialart</th><th>Abnahme</th>
13                </tr>
14              </thead>
15              <tbody>
16                <tr><td>BL-081201-02</td><td>BLUT</td><td>01.12.2008
17                06:34</td></tr>
18            </tbody>
19          </table>
20        </text>
21      ...
22    <entry typeCode="DRIV">
23      <act classCode="ACT" moodCode="EVN">
24        <entryRelationship typeCode="COMP">
25          <procedure classCode="PROC" moodCode="EVN">
26            <participant typeCode="PRD">
27              <participantRole classCode="SPEC">
28                <id extension="BL-080212-02" root
29                ="2.16.840.1.113883.3.933.1.1"/>
30              <playingEntity>
31                <code code="BLD" codeSystem
32                ="2.16.840.1.113883.5.129"
33                codeSystemName="HL7.SpecimenType" displayName="
34                Whole blood"/>
35              ...

```

Die Strukturierung der medizinischen Inhalte wird bei der CDA vorwiegend als Freitext in sogenannten CDA-Implementierungsleitfäden beschrieben. Für die österreichische ELGA existieren momentan fünf CDA-Implementierungsleitfäden [4]. Im allgemeinen ELGA-CDA-Implementierungsleitfäden werden grundlegende Regeln für alle anderen ELGA-CDA-Implementierungsleitfäden festgesetzt und die Information im *Header* beschrieben. In den ELGA-CDA-Implementierungsleitfäden für Entlassungsbrief ärztlich, Entlassungsbrief Pflege, Laborbefund und Befund bildgebende Diagnostik können diese Einschränkungen noch erweitert werden. Weiters werden dort die medizinischen Inhalte im *Body* für das jeweilige Anwendungsgebiet eingeschränkt. Alternativ zu den Implementierungsleitfäden existiert ein “Vorstandard” für CDA-Templates [30]. CDA-Templates haben denselben Fokus wie Archetypen und erlauben es, das Wissen in Implementierungsleitfäden in computerverarbeitbarer Form darzustellen. Im Rahmen der vorliegenden Arbeit konnten jedoch keine Implementierungen gefunden werden, die auf dem Template-Modell basieren.

Im Gegensatz zur CDA und dem Standard ISO/EN 13606 ist die openEHR-Spezifikation nicht als offizieller Standard veröffentlicht. Dies hat den Vorteil, dass Änderungen schneller in die Spezifikation übernommen werden können. Der Nachteil ist die verringerte Investitionssicherheit, da Änderungen in der Spezifikation zu Anpassungen in der Software führen. Die openEHR-Spezifikation kann als eine Übermenge des Standards ISO/EN 13606 gesehen werden, d.h. ISO/EN-13606-EHR-Extrakte können ohne Informationsverlust in openEHR-Dokumente umgewandelt werden, aber nicht vice versa. Das Konzept der openEHR-Archetypen wurde fast ident in die Archetypenspezifikation im Standard ISO/EN 13606 übernommen. In der aktuellen Version von openEHR wird zusätzlich das Konzept der openEHR-Templates [48] eingeführt. Im Gegensatz zu den CDA-Templates, die mit Archetypen vergleichbar sind, dienen openEHR-Templates dazu, Archetypen für die Nutzung in konkreten Anwendungsfällen weiter einzuschränken. Archetypen können so zum Beispiel für die Visualisierung als Formulare zusammengefasst werden oder optionale Felder ausgeblendet werden. Die openEHR-Spezifikation beschränkt sich nicht wie die CDA auf den Austausch von Dokumenten, sondern kann auch direkt als internes Datenmodell von Informationssystemen verwendet werden. Neben der openEHR-Spezifikation bietet openEHR zusätzlich Open-Source-Software und Tools basierend auf der Spezifikation an. Die openEHR-Spezifikation wurde auch bereits in kommerziellen Produkten umgesetzt [49].

Die ISO/EN-13606-Archetyphen bauen auf einer älteren Spezifikation der openEHR-Archetyphen auf. Trotz desselben zugrundeliegenden Archetyphenmodells haben die zwei Arten von Archetyphen eine unterschiedliche Ausrichtung. ISO/EN-13606-Archetyphen dienen in erster Linie dazu, medizinische Inhalte zu beschreiben, die mittels EHR-Extrakten ausgetauscht werden. openEHR-Archetyphen werden im Gegensatz dazu nicht nur für die Beschreibung des Austauschformats verwendet, sondern ebenfalls zur Beschreibung von in Routinesystemen erhobenen Daten. Um die dadurch bedingten spezifischeren Eingabemöglichkeiten besser abdecken zu können, werden in der openEHR-Welt tendenziell mehr Archetyphen zur Beschreibung eines medizinischen Konzepts verwendet, als dies beim Standard ISO/EN 13606 der Fall ist.

Unter *Detailed clinical models* (DCM) [50] werden neben Archetyphen und HL7-Templates weitere Konzepte zusammengefasst, die es erlauben medizinisches Wissen zu formalisieren.

### **2.1.3 Archetyphen und EHRs**

Im Folgenden werden Ansätze vorgestellt, die das Erzeugen von EHR-Dokumenten gemäß den Vorgaben von Archetyphen ermöglichen, das Validieren von EHR-Dokumenten erlauben sowie Ansätze zum Verwalten von Archetyphen aufgezeigt.

#### **2.1.3.1 Erzeugen standardisierter, archetyphenkonformer EHR-Dokumente**

Der durch den medizinischen Fortschritt bedingte laufende Wissenszuwachs erfordert eine laufende Adaptierung von EHR-Systemen. Die automatische Generierung von Formularen aus Archetyphen ermöglicht es EHR-Systemen, vorher nicht bekannte medizinische Inhalte „on-the-fly“ darzustellen und zu editieren. Im Folgenden werden Ansätze zum Erzeugen von EHR-Dokumenten, die den Vorgaben von Archetyphen entsprechen, vorgestellt.

In [51] wird ein Ansatz für die automatische Generierung von Formularen aus Archetyphen vorgestellt. Aus Archetyphen werden XML-Beschreibungen in einer XML-basierten GUI-Sprache erzeugt. Die Arbeit zeigt auf, dass in Archetyphen ausreichend Information enthalten ist, um Formulare automatisch zu erzeugen.

Eine automatische Integration von Archetyphen ohne zusätzliche manuelle Arbeit wurde in [52] unter dem Namen *Plug-and-Play-Integration von Archetyphen* vorgeschlagen. Mit dem medizinischen Wissen der Archetyphen werden dabei in einem existierenden System Formulare

automatisch erzeugt. Die Benutzer können mit diesen Formularen medizinische Information basierend auf den Archetypen dokumentieren. Das Wissen, das zum Erzeugen der Formulare aus Archetypen verwendet wurde, kann beim Export der Dokumente herangezogen werden, um Archetypen-konforme Dokumente zu erstellen.

Archetypen können, wie in [53] und [54] beschrieben, in existierende EHR-Systeme integriert werden. Mit diesem Ansatz kann auf eine bestehende Infrastruktur aufgebaut werden, d.h. Rechteverwaltung, Dokumentenmanagement etc. müssen nicht neu implementiert werden. Bei der Entwicklung von Prototypen müssen jedoch die Eigenschaften des existierenden Systems berücksichtigt werden, was wiederum zu einem erhöhten Entwicklungsaufwand führt.

Die im Folgenden beschriebenen EHR-Systeme bauen nicht auf bestehenden Systemen auf, sondern sind speziell auf die Anforderungen von Archetypen zugeschnitten. *Opereffa* [55] ist ein openEHR-Referenz-Framework. Es ermöglicht das Generieren von Formularen aus openEHR-Archetypen und das Erstellen von Dokumenten basierend auf diesen Formularen. Ein ähnlicher Ansatz wird in [56] mit dem System *EHR-Flex* vorgestellt. *EHR-Flex* ist referenzmodellunabhängig, unterstützt in der aktuellen Version jedoch nur den Standard ISO/EN 13606. Durch Einführung eines *Template Objekt Modells* soll es aber möglich sein, das Benutzerinterface auch für andere Referenzmodelle anzupassen.

### **2.1.3.2 Semantische Validierung von EHR-Dokumenten**

Es können zwei Typen von Validierung von EHR-Dokumenten, die auf dem Zwei-Modell-Ansatz basieren, unterschieden werden. Erstens die syntaktische Validierung, bei der nur die Vorgaben des Referenzmodells berücksichtigt werden, und zweitens die semantische Validierung, bei der zusätzlich die Vorgaben des Archetyps bzw. der CDA-Implementierungsleitfäden validiert werden. Ein semantisch interoperabler Austausch von EHR-Dokumenten kann nur gewährleistet werden, wenn neben der syntaktischen Validierung auch die semantische Ebene validiert wird. Nur wenn die Vorgaben der Archetypen richtig umgesetzt sind, ist es möglich, eine Suche, Filterung oder Darstellung basierend auf den im Archetyp vorgegebenen Strukturen durchzuführen.

Aktuelle Ansätze zum Validieren von Zwei-Modell-Ansatz-Daten verwenden in der Regel XML-Schema zum Validieren der Vorgaben des Informationsmodells (i.e. syntaktische Validierung) und Schematron-Skripte zum Validieren der Vorgaben der Wissensmodelle (i.e.

semantische Validierung).

Für die Validierung der Vorgaben der CDA-Implementierungsleitfäden werden primär Schematron-Skripte [57] verwendet. Es existieren unterschiedliche Online-Validierungstools [58, 59], die Schematron-Skripte direkt mittels Webservices einbinden. Auf der Homepage von ELGA [4] bzw. der Schweizer HL7-Benutzergruppe [60] können Schematron-Skripte für die nationalen Implementierungen heruntergeladen werden.

In [61] und [62] werden jeweils Implementierungen zum Austausch von ISO/EN-13606-Extrakten vorgestellt. Die erste Implementierung ist ein Kommunikationsserver, der archetypkonforme EHR-Extrakte zwischenspeichert und weiterleitet. Die zweite Implementierung dient dazu, Daten einer Intensivstation an einen EHR-Server weiterzuleiten. In beiden Fällen wird angegeben, dass die EHR-Extrakte mittels XML-Schema validiert werden, ob es sich dabei jedoch nur um eine syntaktische oder auch um eine semantische Validierung handelt, wird nicht näher erläutert. Da keine Angaben bezüglich der *unique particle attribution constraint rule* (siehe Kapitel 3.4) von XML-Schema gemacht werden, ist davon auszugehen, dass nur eine syntaktische Validierung durchgeführt wurde.

Für die semantische Validierung von openEHR-Dokumenten wurde im Jahr 2002 in [63] ein Ansatz basierend auf XML-Schema und XSLT beschrieben. Seit kurzem arbeitet die openEHR Foundation an einer Programmierschnittstelle (engl. *API*) zum semantischen Validieren von openEHR-Dokumenten [64]. Dazu werden die Archetypen geparkt und mittels Java zur Validierung von EHR-Dokumenten verwendet. Die programmatische Umsetzung der *openEHR Java API* befindet sich noch in Entwicklung.

### **2.1.3.3 Archetyp-Repositorys**

Um Archetypen zu organisieren und die Erzeugung von neuen Archetypen zu unterstützen, wurden in den letzten Jahren unterschiedliche Archetyp-Repositorys erstellt. Abhängig vom Forschungsschwerpunkt sind diese unterschiedlich umgesetzt. Im Folgenden werden die unterschiedlichen Ansätze vorgestellt. Es kann davon ausgegangen werden, dass neben den hier präsentierten Archetyp-Repositorys noch einige andere existieren, die etwa in kommerziellen Softwarelösungen in Krankenhäusern zum Einsatz kommen. Da zu diesen keine öffentliche Information vorhanden ist, liegt der Fokus hier auf den öffentlich zugänglichen Archetyp-Repositorys.

Das bekannteste und größte der Öffentlichkeit zugängliche Archetyp-Repository ist der *Clinical Knowledge Manager* [65] von openEHR. Er verfügt über ein Webinterface und beinhaltet momentan 288 openEHR-Archetyphen (262 mit Status Draft, 10 mit Status veröffentlicht, 16 mit Status "teamreview"). Archetyphen können im *Clinical Knowledge Manager* verändert, übersetzt und kommentiert werden. Es existiert ein Versionierungskonzept, bei dem alle alten Versionen eines Archetyp eingesehen werden können. Die *ADL Workbench* [66] von openEHR ermöglicht die Offline-Verwaltung von Archetyphen. Archetyphen, die in die Workbench geladen werden, werden geparkt und als Baumstruktur dargestellt. Es ist hier auch möglich, *Slots* und Verzweigungen von Archetyphen nachzuverfolgen.

In [67] wird das virtuelle Archetyp-Repository *ArchMS* vorgestellt. Zu jedem Archetypen wird die *URL*, unter der der Archetyp gefunden werden kann, sowie das zugrundeliegende Referenzmodell gespeichert. Es werden die Mechanismen des *Semantic Webs* verwendet, dazu werden alle ADL-Archetyphen in ein *OWL*-Format (siehe Kapitel 2.4.3) umgewandelt. Das ermöglicht im nächsten Schritt Archetyphen mit Anmerkungen zu versehen (z.B. Verknüpfungen zu anderen Ontologien), eine semantische Suche von Archetyphen und das Vergleichen von Archetyphen. In [68] wird beschrieben, wie über ein Webinterface Suchanfragen an das *ArchMS*-System und die darin enthaltenen Archetyphen abgesetzt werden. Es werden ISO/EN-13606- sowie openEHR-Archetyphen unterstützt.

Indirekt mit den Archetyp-Repositorys verwandt sind Terminologieserver. In einem Terminologieserver werden Terminologien (kontrollierte Vokabeln, Klassifikationen, etc.) für den Benutzer in maschinenlesbarer Form zur Verfügung gestellt. In [69] wird ein Terminologieserver, der für die Verwendung im Rahmen der österreichischen Elektronischen Gesundheitsakte evaluiert wird, beschrieben. Archetyphen bieten über die *Termbinding Section* die Möglichkeit, diese Terminologien direkt einzubinden. Das über Archetyphen festgelegte kontrollierte Vokabular kann als eigene Terminologie angesehen werden.

Die beiden archetypbasierten Umsetzungen in [70] und [71] basieren auf Semantic-Web-Technologien. Archetyphen, die mit der Archetype Definition Language beschrieben sind, werden dabei in die *RDF-Triple* (siehe Kapitel 2.4.3) umgewandelt.

In [72] werden drei Ontologien vorgestellt, um EHR-Dokumente der unterschiedlichen Standards (i.e. openEHR, HL7 CDA, ISO/EN 13606) mittels eines Ontologie-Mappings in ein EHR-Dokument eines anderen Standards zu transformieren. In CEN-SP werden die

Listing 2.3: Beispiel der \*RDF-Triple\* aus [72]

```
1 <RDF xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" ... >
2 <Ontology about="">
3   <imports resource="http://klt.inf.um.es/~cati/ontologies/CEN-AR-v2.0.owl
4     "/>
5 </Ontology>
6 ...
7 <cen:ELEMENT ID="at0001">
8   <has_occurrence_constraint>
9     <cen-archetype:OCCURRENCE rdf:ID="OCCURRENCE_0_1">
10    <cen-archetype:interval rdf:ID="#INTEGER_INTERVAL_0_1"/>
11  </cen-archetype:OCCURRENCE>
12 </has_occurrence_constraint>
13 <cen_element_value>...</cen_element_value>
14 <term_definitions>...</term_definitions>
15 </cen:ELEMENT>
16 ...
17 </RDF>
```

Datenstrukturen und Datentypen, in CEN-RM alle Beziehungen innerhalb des Referenzmodells und in CEN-AR AOM das Archetype Object Model beschrieben. Jeder einzelne Archetyp wird als Instanz dieser Ontologien mittels *RDF-Triple* abgebildet. Die Ontologien enthalten Konzepte, die in ADL umsetzbar sind, wie Kardinalitäten, aber auch für die Klassen des Referenzmodells wie CLUSTER etc. Wie in Listing 2.3 in Zeile 6 ersichtlich, werden in [72] die Archetyp-Knoten nur innerhalb eines Archetyps eindeutig zugeordnet. Es sind keine global eindeutigen *URI* zugewiesen.

In [73] wird ein Ansatz für das automatische Übersetzen von openEHR-Archetypen in *OWL* beschrieben. Ziel ist es, im Gegensatz zu [70], bestehende Daten auf Archetypen zu mappen bzw. Regeln und Schlussfolgerungen auf archetypbasierte Daten zu ermöglichen. Der Java-Code zum Übersetzen von ADL in *OWL* wird unter einer Open-Source-Lizenz zur Verfügung gestellt.

## 2.2 XML

Die *Extensible Markup Language* (XML) ist eine Technologie, die sich mit der Beschreibung und Strukturierung von Daten beschäftigt. Um die Vorteile von Binärdateien (i.e. vom Computer interpretierbar, kompakt, mit der Möglichkeit, Metadaten einzubinden) und Textdateien (i.e. universell austauschbar) zu kombinieren, wurde die *Standard Generalized*

*Markup Language* (SGML) entwickelt. SGML ist eine textbasierte Sprache, die es erlaubt, Metadaten zu Daten "selbstbeschreibend" hinzuzufügen [74]. XML ist ein Subset von SGML, in dem die Komplexität von SGML verringert wurde. XML ermöglicht es, hierarchische Strukturen in textueller Form darzustellen.

Ein XML-Dokument ist *wohlgeformt*, wenn es alle Regeln von XML erfüllt. Ein XML-Dokument muss immer genau ein Wurzelement besitzen. Das Wurzelement und alle weiteren Elemente müssen immer aus einem Start- und einem End-Tag bestehen (z.B. `<tag>1</tag>` bzw. `<tag/>` für das leere Element). Die Elemente müssen immer in der gleichen hierarchischen Ebene geöffnet und geschlossen werden und es darf zu keiner Überlappung von Elementen kommen. Elemente können beliebig viele Attribute enthalten (z.B. `<tag name='1'>`), wobei ein Attributname pro Element nur einmal verwendet werden darf. Kommentare können in XML begrenzt durch die *Keywords* `<!--` und `-->` formuliert werden.

Zusätzlich zu der *Wohlgeformtheit* können noch weitere Einschränkungen mittels *Document Type Definitions* (DTDs) bzw. *XML-Schemas* [75] gemacht werden. Jede Einschränkung von DTD kann auch mittels XML-Schema gemacht werden, aber nicht umgekehrt. Im Rahmen der vorliegenden Arbeit wird nur XML-Schema verwendet. Ein XML-Schema ist ein XML-Dokument, in dem die Struktur eines anderen XML-Dokuments beschrieben wird. XML-Schema kann somit für die Validierung von XML-Dokumenten verwendet werden. XML-Schema bietet eigene Datentypen an, die für das Beschreiben von XML-Strukturen verwendet werden können und zum Beispiel für eine Optimierung der Suchen in XML-Dokumenten herangezogen werden können.

Das Adjektiv *Extensible* in XML bedeutet, dass Elemente frei definiert werden können. Um Elemente zu unterscheiden, die zwar denselben Namen, aber eine unterschiedliche Bedeutung haben, da sie zum Beispiel von unterschiedlichen Organisationen definiert wurden, werden *Namespaces* verwendet. Ein *Namespace* wird über das Attribut "xmlns", einen Präfix sowie über eine URL definiert, über die der *Namespace* eindeutig zuordenbar ist. Im Falle des *default Namespace* wird kein Präfix angegeben. Ein Beispiel für eine Namespace-Definition ist in Listing 2.4 in Zeile 2 zu sehen. Hier sind die *Namespaces* im Wurzelement definiert worden, sie können aber auch in allen anderen Elementen definiert werden. Der *Namespace* ist jeweils für alle Kinder-Elemente gültig.

Listing 2.4: Vereinfachte Darstellung eines ISO/EN-13606-EHR-Extrakts in XML

```
1 <?xml version="1.0" encoding="iso-8859-1"?>
2 <EHR_EXTRACT xmlns="http://www.en13606.org/RM" xmlns:xsi="http://www.w3.
   org/2001/XMLSchema-instance">
3 <all_compositions xsi:type="COMPOSITION">
4 <name xsi:type="SIMPLE_TEXT">
5 <originalText>Arztbrief vom Internisten</originalText>
6 </name>
7 <content xsi:type="SECTION">
8 <name xsi:type="SIMPLE_TEXT">
9 <originalText>Diabetes Klassifikation</originalText>
10 </name>
11 <members xsi:type="ENTRY">
12 <name xsi:type="SIMPLE_TEXT">
13 <originalText>Diabetes Klassifikation</originalText>
14 </name>
15 <items xsi:type="ELEMENT">
16 <name xsi:type="SIMPLE_TEXT">
17 <originalText>DM Erstdiagnose</originalText>
18 </name>
19 <value xsi:type="DATE">
20 <date>1995-02-01</date>
21 </value>
22 </items>
23 </members>
24 </content>
25 </all_compositions>
26 </EHR_EXTRACT>
```

Bei XML-Dokumenten kann zwischen *dokumentzentriert* und *datenzentriert* unterschieden werden [76]. Dokumentzentrierte XML-Dokumente sind hauptsächlich für den menschlichen Gebrauch gedacht. Die XML-Struktur dient dazu, die Struktur des Textes abzubilden. Der Inhalt eines Level-2-CDA-Dokuments mit seinen Elementen wie `<section>` für Absätze ist ein Vertreter eines dokumentzentrierten XML-Dokuments (siehe Listing 2.2 auf Seite 23 Zeile 7–16). Im Gegensatz dazu beinhalten datenzentrierte XML-Dokumente feingranuläre XML-Strukturen, die in der Regel aus Datenbanken generiert werden und einzelne Werte und Ausprägungen in eigenen XML-Elementen oder Attributen abbilden. Listing 2.4 zeigt ein datenzentriertes XML-Dokument. Diese Unterscheidung hat Auswirkungen auf die Suche in XML-Dokumenten. In dokumentzentrierten XML-Dokumenten können einzelne Werte unter anderem nur mithilfe einer Freitextsuche gefunden werden, in datenzentrierten

XML-Dokumenten ist es möglich, sofern die Struktur der Daten bekannt ist, gezielt einzelne Werte zu suchen und für eine automatische Weiterverarbeitung zu verwenden. Diese Arbeit beschäftigt sich mit datenzentrierten XML-Dokumenten.

Um XML-Dokumente in einer Applikation zu verarbeiten, müssen diese geparkt werden. Für das Parsen benötigte Informationen werden am Anfang eines XML-Dokumentes angegeben. In der ersten Zeile in Listing 2.4 ist die XML-Deklaration zu sehen. Sie gibt an, dass es sich bei diesem Dokument um ein XML-Dokument handelt. Das Versions-Attribut gibt an, um welche Version von XML es sich handelt (i.e. 1.0). Da XML-Dokumente Textdateien sind, muss die Codierung des Textes berücksichtigt werden. XML-Dokumente werden standardmäßig als *UTF-8* codiert; in dieser Arbeit wird durchgehend *ISO-8859-1* als Codierung verwendet.

Die zwei am weitesten verbreiteten Technologien, um XML-Dokumente zu parsen, sind das *Document Object Model* (DOM) und die *Simple API for XML* (SAX). DOM-Parser stellen das gesamte Dokument als Baumstruktur zur Verfügung und der Anwender kann in diesem Baum Knoten auswählen und zu anderen Knoten navigieren. DOM-Bäume sind leicht zu bearbeiten, haben aber den Nachteil, dass immer das gesamte XML-Dokument geparkt werden muss, was bei einem großen Dokument zu einem großen Datenverbrauch und Performanceeinschränkungen führen kann. Ein SAX-Parser ist Event-getrieben. Schon beim Einlesen des ersten XML-Elements kann mit der Verarbeitung begonnen werden. Dies hat den Vorteil, dass nur die aktuell benötigte Information im Speicher vorhanden ist. SAX wird verwendet, wenn ein XML-Dokument nur einmal verarbeitet wird, um beispielsweise eine Suche oder eine Transformation durchzuführen.

Die Grundlage für die Transformation oder Suche von XML-Dokumenten bildet die *XML Path Language* (XPath) [77]. XPath ist eine Abfragesprache für XML-Dokumente, die es erlaubt, Teile eines XML-Dokuments auszuwählen bzw. im XML-Dokument zu navigieren. Ein XPath-Ausdruck wird aus unterschiedlichen *Lokalisierungsschritten* zusammengesetzt, die jeweils durch einen Schrägstrich (i.e. “/”) getrennt werden. Ein *Lokalisierungsschritt* kann aus Achsen (engl. *axis*), Knotentest (engl. *node-test*) und Prädikaten (engl. *predicate*) bestehen. Über die Achsen können Kinderknoten, Elternknoten, Geschwisterknoten etc. referenziert werden. Über den Knotentest werden die auszuwählenden Elemente einer Achse zusätzlich eingeschränkt. Prädikate werden in eckigen Klammern angegeben und erlauben es, die Rückgabewerte weiter einzuschränken. Es stehen dabei unter anderem Funktionen,

arithmetische Ausdrücke und logische Ausdrücke zur Verfügung. Es werden jene Teile beim Ausführen eines XPath-Ausdrucks zurückgegeben, die den im XPath-Ausdruck angegebenen Werten entsprechen.

*XSL Transformations* (XSLT) [78] ist eine Programmiersprache in XML, die auf der *Extensible Stylesheet Language* (XSL) basiert. ISO/EN-13606-EHR-Extrakte können mit sogenannten XSLT-Stylesheets in ein anderes Ausgabeformat transformiert werden. Im Web wird XSLT oft zum Erzeugen von HTML-Dokumenten aus XML-Dokumenten verwendet, es ist aber prinzipiell auch jedes andere Ausgabeformat möglich. In XSLT-Skripts werden XPath-Ausdrücke verwendet, um die relevanten Teile für die Transformation zu finden. In der vorliegenden Arbeit wird XSLT verwendet, um ISO/EN-13606-EHR-Extrakte im Browser darzustellen. Weiters kann eine XSLT-Transformation auch als einfache Suche in einem Dokument angesehen werden, indem das XSLT-Skript die Suchanfrage abbildet und das Ausgabedokument die gewünschten Suchergebnisse beinhaltet.

Während bei XSLT der Fokus auf der Transformation von XML-Dokumenten liegt, wird die *XML Query Language* (XQuery) [79] verwendet, um Daten in einer Sammlung von XML-Dokumenten zu suchen. XQuery-Ausdrücke können im *FLOWR*-Format (Abkürzung für *FOR*, *LET*, *ORDER BY*, *WHERE* und *RETURN*) dargestellt werden, siehe Listing 2.5. Der Aufbau einer XQuery ist vergleichbar mit SQL-Ausdrücken, die in relationalen Datenbanken für Suchanfragen verwendet werden. *FLOWR*-Ausdrücke können beliebig geschachtelt werden, um verschachtelte UND/ODER-Suchanfragen, wie sie später in der vorliegenden Arbeit verwendet werden, abzubilden. In [76] werden *tag-based*-, *path-based*- und *clause-based*-Sprachen unterschieden. XPath ist eine *path-based*-Sprache, XQuery eine *clause-based*-Sprache.

Listing 2.5: Beispiel einer einfachen *FLOWR*-XQuery-Abfrage

```
1 xquery version "1.0";
2 for $d in //*[archetype_id='CEN-EN13606-CLUSTER.blood_pressure.v1/at0000']
3 let $e := $d/parts
4 where $e/archetype_id = 'at0001'
5     and $e/value/value > 149
6 return <result>{$e}</result>
```

Die Verarbeitung von XQuery-Abfragen wird über einen XQuery-Prozessor auf XML-Dokumente angewendet. XML-Datenbanken bieten eigene Implementierungen von

XQuery-Prozessoren an, in dieser Arbeit wird *Saxon* [80] verwendet. *Saxon* ist ein Open Source XSLT und XQuery-Prozessor, der als JAVA- und .NET-Implementierung bereitgestellt wird. *Saxon* wird in einer kommerziellen und einer freien Version zur Verfügung gestellt. Die kommerzielle Version bietet zusätzlich die Möglichkeit, XML-Schema beim Ausführen einer XQuery miteinzubeziehen, um die XQuery auf Gültigkeit zu überprüfen und das Debuggen zu verbessern [74].

Die in dieser Arbeit verwendeten *Semantic-Web-Technologien* (siehe Kapitel 2.4.3) und die Transaktionen zwischen den verwendeten Shared-EHR-Systemen (siehe Kapitel 2.3) basieren ebenfalls auf XML.

## 2.3 Shared-EHR-System-Architekturen

Es existieren unterschiedliche Ansätze, wie medizinische Daten zwischen GDAs ausgetauscht werden können. Im Standard ISO/TR 20514 [12] werden lokale EHR-Systeme und Shared-EHR-Systeme unterschieden. In [81] werden die unterschiedlichen Shared-EHR-System-Architekturen weiter in folgende Kategorien unterteilt.

- Bei der zentralen Datenhaltung werden alle Daten auf einem zentralen Server gespeichert. Dies setzt eine ständige Datenverbindung zum Server voraus, alle beteiligten GDAs können jedoch auf alle Daten zugreifen. Die Missbrauchsgefahr ist hier hoch, da alle Daten zentral gespeichert sind.
- Bei der dezentralen Datenhaltung werden die Daten lokal beim GDA gespeichert, über einen Kommunikationsserver werden die Daten bei Bedarf zu den anderen GDAs geschickt. Eine zentrale Suche ist nicht möglich, Dokumente werden von einem GDA zum anderen gesendet.
- Bei der mobilen Datenhaltung werden die Daten auf einem mobilen Datenträger (z.B. USB-Stick) gespeichert und vom Patienten verwaltet. Der Vorteil der geringen Missbrauchsgefahr wird dadurch verringert, dass bei Verlust des Datenträgers alle Daten verloren gehen.
- Eine Mischform ist die dezentrale Datenhaltung mit zentraler Metadatenkomponente. Die dezentrale Datenhaltung hat den Vorteil einer geringeren Missbrauchsgefahr, eine Suche

nach relevanten Dokumenten wird dadurch jedoch erschwert. Durch die Kombination mit einer zentralen Metadatenkomponente können über die Metadaten Informationen gezielt gesucht werden und relevante Dokumente angefordert werden.

Diese Arbeit konzentriert sich im Folgenden auf Shared-EHR-System-Architekturen mit dezentraler Datenhaltung und zentraler Metadatenkomponente. In Abbildung 2.4 werden die Akteure einer Shared-EHR-System-Architektur mit dezentraler Datenhaltung und zentraler Metadatenkomponente gezeigt. Die standardisierten EHR-Dokumente werden von einem GDA erstellt und in seinem System dezentral abgelegt, die Metadaten der neuen EHR-Dokumente werden an die zentrale Metadatenkomponente gesendet und dort registriert. Ein GDA kann über die Metadatenkomponente EHR-Dokumente der anderen GDAs suchen und diese direkt von deren Systemen herunterladen.

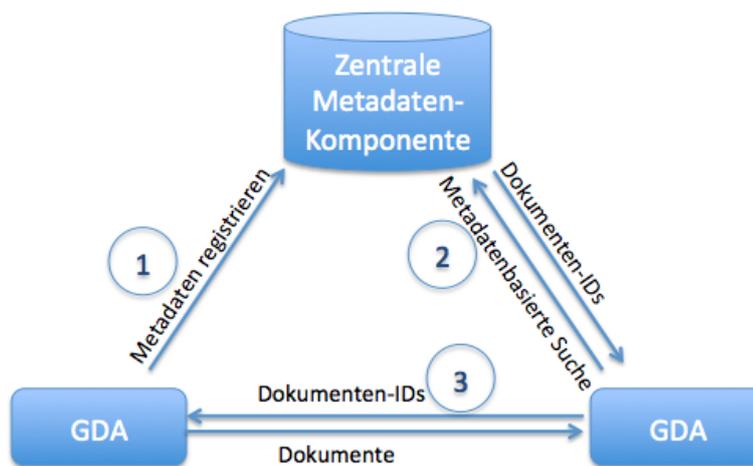


Abbildung 2.4: Dezentrale Datenhaltung mit zentraler Metadatenkomponente. 1) Ein GDA registriert neue Dokumente mit den geforderten Metadaten in der zentralen Metadaten-Komponente, 2) über eine metadatenbasierte Suchanfrage werden die Dokumenten-IDs der relevanten Dokumente gesucht und 3) die relevanten Dokumente direkt vom jeweiligen GDA heruntergeladen.

Die dezentrale Speicherung der Dokumente hat den Vorteil, dass Datenmissbrauch verhindert wird. Um diesen Vorteil nicht zu verlieren, darf jedoch kein zentraler Index der Inhalte der EHR-Dokumente angelegt werden. Dies hat zur Folge, dass in dieser Architektur nur eine direkte Suche nach Metadaten möglich ist.

### 2.3.1 Das IHE-Profil Cross Enterprise Document Sharing (XDS)

*Integrating the Healthcare Enterprise* (IHE) ist eine internationale Organisation zur Förderung und Entwicklung und Harmonisierung von Spezifikationen für den Datenaustausch zwischen Informationssystemen im Gesundheitswesen mit Fokus auf Interoperabilität. Basierend auf Anforderungen aus der Praxis werden existierende Standards (i.e. ISO, CEN, HL7, W3C Recommendations, DICOM etc.) gesucht daraus sogenannte *Profile* definiert. Auf *Connectathons* können die umgesetzten Profile von Herstellern untereinander getestet werden. Wird ein IHE-Profil umgesetzt, wird es in ein Produktregister aufgenommen, in dem Anwender Produkte finden können, die die Anforderungen erfüllen. Ein Überblick über die Vorgangsweise von IHE ist in Abbildung 2.5 zu sehen.

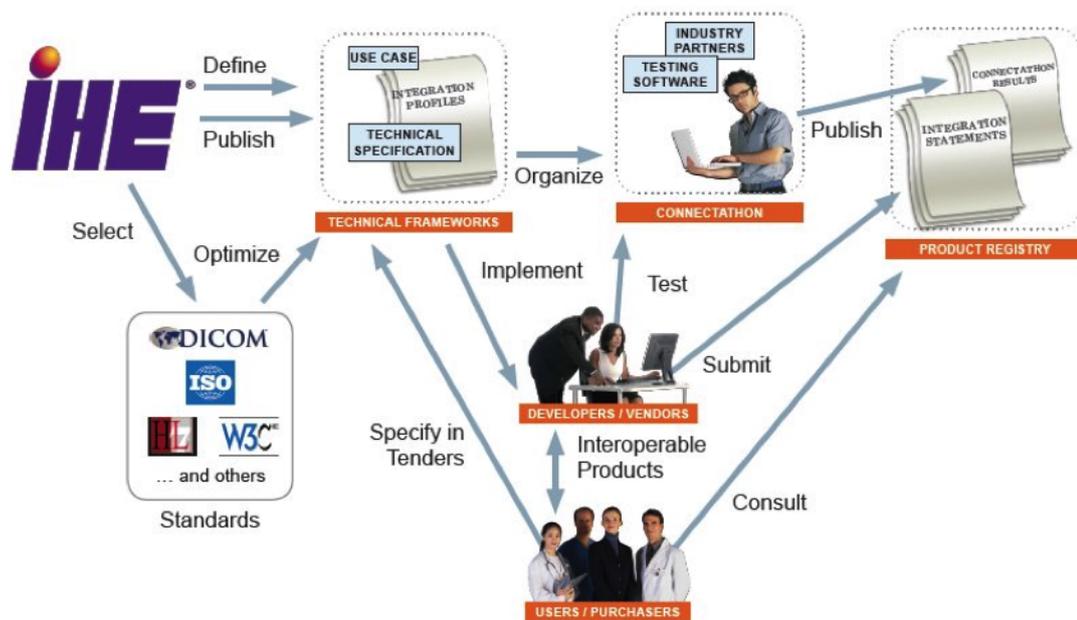


Abbildung 2.5: Überblick über die zentralen Aktivitäten von IHE (Grafik von [www.ihe.net](http://www.ihe.net)).

In einem IHE-Profil werden die beteiligten Akteure und deren Transaktionen beschrieben und unterschiedlichen Domänen zugeteilt, die ein zusammenhängendes medizinisches Fachgebiet abgrenzen (z.B. Pathologie, Augenheilkunde, IT-Infrastruktur, Labor, Radiologie, etc.). Weiters können die IHE-Profile in eine von drei Kategorien eingeteilt werden:

- fertige IHE-Profile

- IHE-Profile in einer Testphase
- zurückgezogene IHE-Profile

Eine Auflistung aller aktuellen IHE-Profile kann unter [82] gefunden werden.

Das IHE-Profil *Cross Enterprise Document Sharing* (XDS) [16] ist eine Shared-EHR-System-Architektur mit dezentraler Datenhaltung und zentraler Metadatenkomponente und befindet sich in der Domäne IT-Infrastruktur. IHE-XDS ist eine Spezifikation für den Austausch von medizinischen Daten unabhängig von deren Inhalt innerhalb einer sogenannten *XDS Affinity Domain*. Eine *XDS Affinity Domain* besteht aus einer Gruppe von GDAs, die sich auf eine gemeinsame Strategie zum Datenaustausch geeinigt haben (z.B. regionales oder nationales Gesundheitsnetz, Zusammenschluss von Krankenhäusern etc.). Wenn nicht explizit anders angegeben, wird im Folgenden mit IHE-XDS die aktuelle Version XDS.b referenziert.

IHE-XDS gewinnt in regionalen und nationalen Shared-EHR-Systemen immer mehr an Bedeutung. Der Blogeintrag [83] mit dazugehöriger Google Map gibt einen Überblick, wo in der Welt IHE-XDS überall eingesetzt wird. Einige Implementierungen werden hier kurz beschrieben. In der südafrikanischen Provinz Gauteng wird ein System basierend auf IHE-XDS realisiert, das 10 Millionen Patienten Zugriff auf ihre Gesundheitsdaten ermöglichen soll [84]. Im Schweizer Kanton Sankt Gallen wurde für Kardiologen ein System für eine geteilte oder verteilte Versorgung umgesetzt [85]. Durch die IHE-XDS wird eine gemeinsame Diagnose und Behandlung eines Patienten ermöglicht. In der italienischen Provinz Venetien werden klinische Dokumente von 4,9 Millionen Patienten mittels IHE-XDS zur Verfügung gestellt [86]. Auch in Frankreich wird das nationale Shared-EHR-System (*Dossier Medical Personnel* kurz DMP) mit IHE-XDS umgesetzt [87].

In Österreich wurde im November 2012 die Umsetzung eines nationalen Shared-EHR-System, das ebenfalls auf der IHE-XDS-Architektur aufbauen soll, beschlossen. IHE-XDS wird jetzt schon im niederösterreichischen Gesundheitsnetz mit 2 Millionen Patienten und 27 Krankenhäusern [88], sowie im Gesundheitsnetz Tirol für den Datenaustausch eingesetzt [89]. Die ausgetauschten Daten basieren bei den österreichischen Umsetzungen auf der CDA.

Abbildung 2.6 zeigt einen Überblick über die Akteure und Transaktionen, die an IHE-XDS beteiligt sind. Ein *Document Source*-Akteur sendet ein Dokument an den *Document*

*Repository*-Akteur. Dabei werden spezielle Metadaten zum Dokument mitgesendet, die es im nächsten Schritt ermöglichen, das Dokument im *Document Registry*-Akteur zu registrieren. Im *Patient Identity Source*-Akteur wird eine eindeutige ID zu jedem Patienten, zu dem Dokumente abgelegt sind, gespeichert. Ein *Document Consumer* kann die registrierten Dokumente in der *Document Registry* suchen und vom *Document Repository* herunterladen.

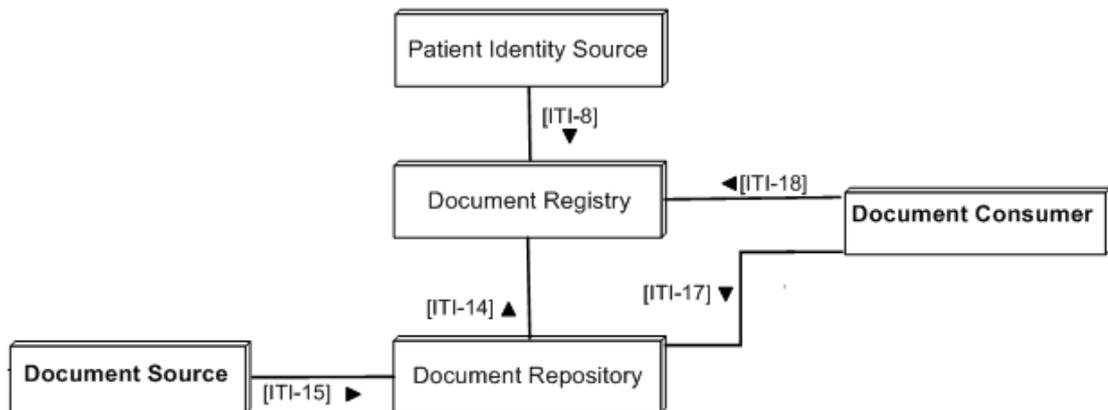


Abbildung 2.6: Überblick der Akteure und Transaktionen in IHE-XDS (aus [16]).

Die Transaktionen bzw. Web-Service-Aufrufe in IHE-XDS werden mit dem Kürzel *ITI* und einer Nummer beschrieben. Die Transaktion *ITI-14* beschreibt die Transaktion *Register Document Set*, *ITI-15* die Transaktion *Provide and Register Document Set*, *ITI-17* die Transaktion *Retrieve Document*, die Transaktion *ITI-18* *Registry Stored Query* und die Transaktion *ITI-8* *Patient Identity Feed*.

Ein wichtiges Konzept in IHE-XDS sind die Dokumenttypen. Jedes Dokument, das in das *Document Repository* hochgeladen wird, muss einem bestimmten Dokumenttyp entsprechen. Dokumenttypen werden zum Beispiel in CDA-Implementierungsleitfäden beschrieben und können verschiedene Arten von Arztbriefen, Befunden etc. sein, wobei für jeden Dokumenttyp die jeweiligen Inhalte vorgegeben sind. Da in XDS Dokumente unabhängig von ihrem Inhalt abgespeichert werden, sofern die Metadaten zur Verfügung stehen, können Dokumente mit unterschiedlichen Dokumentenformaten verwendet werden. Dokumentenformate können zum Beispiel PDF, CDA oder ISO/EN-13606-EHR-Extrakt sein.

### 2.3.2 Ablauf der Suche nach EHR-Dokumenten in IHE-XDS

In IHE existieren unterschiedliche Profile für Suchanfragen zu einem Patienten bzw. einem Patientenkollektiv. Für wissenschaftliche Zwecke ist auch eine patientenübergreifende Suche interessant, um zum Beispiel geeignete Patienten für eine medizinische Studie zu rekrutieren. Der primäre Fokus von IHE-XDS ist es, den GDAs während der Behandlung eines Patienten relevante Dokumente zur Verfügung zu stellen. Alle Suchanfragen beziehen sich in IHE-XDS daher immer auf genau einen Patienten. Die vorliegende Arbeit beschäftigt sich mit der patientenbezogenen Suche.

In einer IHE-XDS-Umgebung werden patientenbezogene Suchanfragen vom Akteur *Document Consumer* an den Akteur *Document Registry* gesendet. Die *Document Registry* beinhaltet Metadaten sowie die genaue Adresse zu den Dokumenten, die dezentral in *Document Repositories* abgelegt sind. Die Transaktion *Registry Stored Query* (ITI-18) (siehe Abbildung 2.6) erlaubt es, für einen Patienten Dokumente nach Dokumenten-Datum (kann ein Zeit-Intervall sein), Dokumenten-Typ (Arztbrief, Laborbefund, etc.), Funktionscodes laut *Leistungsorientierte Krankenanstaltenfinanzierung* (LKF) auf Abteilungsebene (i.e. Code der Organisationseinheit) oder Autoren zu suchen. Unabhängig von der Suche nach Dokumenten eines Patienten sieht die Transaktion *ITI-18* unter anderem die Möglichkeit vor, alle Dokumente eines bestimmten Ursprungs (i.e. *Document Source*), alle Dokumente eines Ordners oder nach dem Zeitpunkt der Übertragung zu suchen. Für die inhaltliche Suche ist jedoch nur der patientenbezogene Zugang relevant. Eine mögliche metadatenbasierte Suchmaske eines *Document Consumers* ist in Abbildung 2.7 zu sehen.

Bei der Transaktion *ITI-18* können nur Suchanfragen gesendet werden, die schon in der *Document Registry* gespeichert sind. Dadurch soll vermieden werden, dass unerwünschter Code über SQL-Anweisungen eingeschleust wird. Außerdem soll dadurch eine Architektur der *Document Registry* ermöglicht werden, die unabhängig von der SQL-Anfrage ist. Es werden nur IDs, die auf eine Suchfunktion in der *Document Registry* verweisen, sowie die Parameter der Suchfunktion übergeben (z.B. für die Suchfunktion *FindDocuments* wird die ID *urn:uuid:14d4debf-8f97-4251-9a74-a90016b0af0d* sowie die Parameter *Patienten-ID*, *Zeitintervall* etc. übergeben).

Jede der oben angeführten Suchen mittels der Transaktion *ITI-18* liefert als Rückgabe eine Liste mit Referenzen auf die Dokumente, die den gesuchten Metadaten entsprechen. Die

Standard Dokumentensuche

**Zeitliche Angaben**

alle

Zeitbereich

Erstellungsdatum:

Zeitbereich:

**Medizinische Organisation**

<input type="checkbox"/> Name des Fachgebietes
<input type="checkbox"/> Augenheilkunde
<input type="checkbox"/> Frauenheilkunde & Geburtshilfe
<input type="checkbox"/> Haut- & Geschlechtskrankheiten
<input type="checkbox"/> Innere Medizin
<input type="checkbox"/> Interdisziplinärer Bereich
<input type="checkbox"/> Medizinisch-chemische Labordiagnostik
<input type="checkbox"/> Neurologie
<input type="checkbox"/> Psychiatrie
<input type="checkbox"/> Andere

**Dokumenteninformation**

<input type="checkbox"/> Dokumententyp
<input type="checkbox"/> Arztbrief
<input type="checkbox"/> Konsil
<input type="checkbox"/> Medizinische Aufnahme
<input type="checkbox"/> Laborbefund
<input type="checkbox"/> Patientendokumentation

Abbildung 2.7: Suchfenster für eine Standard-IHE-XDS-Suchanfrage in einem *Document Consumer*.

jeweiligen Dokumente können dann vom jeweiligen *Document Repository* heruntergeladen werden. In Abbildung 2.8 ist ein Ausschnitt aus dem *Document Consumer* zu sehen, in dem die Metadaten der gefundenen Dokumente als Liste angezeigt werden. Erst beim Anklicken eines Dokuments in dieser Liste wird das jeweilige Dokument aus dem *Document Repository* heruntergeladen.

### 2.3.3 ISO/EN 13606 und IHE-XDS

Die meisten Implementierungen von IHE-XDS verwenden CDA als Dokumentenformat für die Dokumente, die im *Document Repository* abgelegt werden. Aus dem CDA-Header können dabei die von der *Document Registry* benötigten Informationen zum Indizieren der Dokumente direkt heruntergeladen werden.

Die ISO/EN 13606 kann ebenfalls als Dokumentenformat verwendet werden. Die ISO/EN 13606 bildet im Gegensatz zur CDA nicht ein einzelnes Dokument ab, sondern die ganze Gesundheitsakte eines Patienten. In [26] werden zwei Einschränkungen, die dadurch entstehen, angeführt:

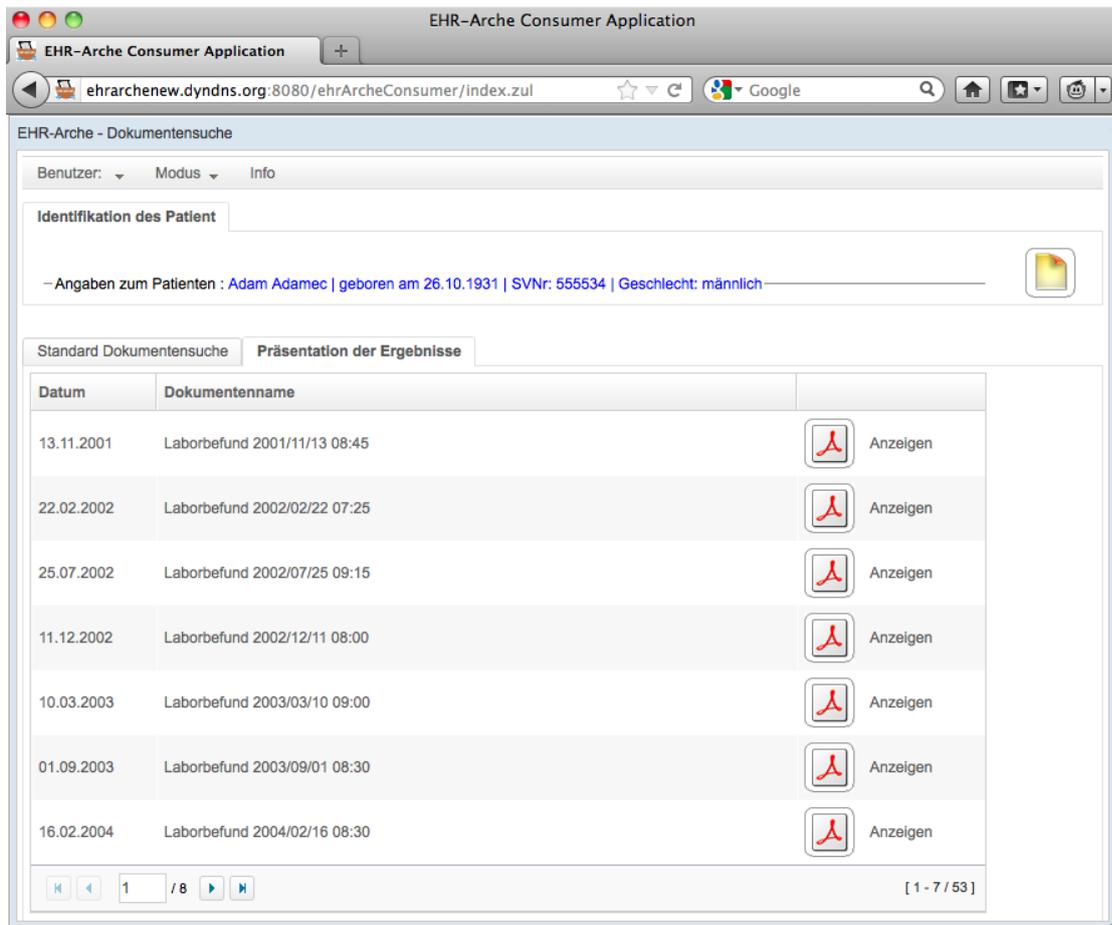


Abbildung 2.8: Suchergebnis einer Standard-IHE-XDS-Suche im *Document Consumer*.

- Es kann nur eine Ordnerhierarchie (*FOLDER*-Klasse) in IHE-XDS abgebildet werden. Weiters können erst über der Hierarchieebene *Folder* Daten abgelegt werden.
- Es können COMPOSITION-Instanzen in IHE-XDS abgelegt werden, wobei deren Inhalt mittels IHE-XDS nicht durchsucht werden kann.

In [90] werden die CDA, die openEHR-Spezifikation sowie die ISO/EN 13606 verglichen und die Möglichkeiten, Instanzen der unterschiedlichen Standards ineinander zu transformieren, analysiert. Mithilfe der CDA-Klassen, die für die Transformation verwendet werden, den IHE-XDS-Metadaten zur Registrierung der CDA-Dokumente, die auf [4] veröffentlicht sind,

sowie dem Mapping der wichtigsten IHE-XDS-Attribute auf Klassen und Attribute des ISO/EN-13606-Referenzmodells in [26] können auch ISO/EN-13606-EHR-Extrakte mit IHE-XDS verwendet werden. Die meisten Metadaten sind dabei in der Klasse EHR\_EXTRACT beschrieben. In [91] wird gezeigt, wie ISO/EN-13606-EHR-Extrakte in ein existierendes *Document Repository* hochgeladen werden. Dabei werden EHR-Extrakte auf COMPOSITION-Ebene und die kommerzielle IHE-XDS-Implementierung *SENSE* der Firma ITH-Icoserve [92] verwendet.

## 2.4 Suchen in EHRs

Die Suche in Shared-EHR-Systemen, im Speziellen der IHE-XDS-Architektur, soll in erster Linie dazu dienen, den behandelnden GDAs zur richtigen Zeit die richtige Information zum Behandeln des Patienten zurückzuliefern, ohne sie mit nicht relevanter Information zu überfluten. Das Vermeiden von Informationsüberflutung spielt in viele Teilbereiche der Informatik hinein. Die Suche im Internet und somit das Vermeiden von Informationsüberflutung wurde durch Google revolutioniert [93]. Im Folgenden wird auf die unterschiedlichen Suchmöglichkeiten eingegangen und die Unterschiede von der Suche im Web und der Suche in Shared-EHR-Systemen erläutert.

Das Forschungsgebiet *Information Retrieval* (IR) (engl. für Informationsbeschaffung) beschäftigt sich mit der computergestützten Suche. Die gewünschte Information wird dabei ausgehend von den Informationsbedürfnissen in Datenbeständen gesucht. Eine Websuche (i.e. Google-Suche) fällt auch unter IR. Für die Suche im Web liegt der Fokus des IR hauptsächlich auf der Suche in unstrukturierten Daten und dem Ranking der Resultate. Die Kombination der Methoden zur Suche in hochstrukturierten Daten in relationalen Datenbanken und unstrukturierten Daten mittels IR kann zur Verbesserung der Suche in XML-Inhalten verwendet werden [94].

### 2.4.1 Arten von Suchanfragen

Abhängig von den Anforderungen an eine Suche können unterschiedliche Arten von Suchanfragen unterschieden werden. In [95] werden unter anderem boolesche Suchanfragen, natürlichsprachliche Suchanfragen, thesaurusgestützte Suchanfragen und Begriff-Suchanfragen unterschieden.

Bei booleschen Suchanfragen (engl. *boolean query*) werden die zu suchenden Ausdrücke mit UND, ODER und NICHT miteinander verknüpft (siehe Abbildung 2.9). Bei einer booleschen Suchanfrage ist ein Wert entweder vorhanden oder nicht, eine Gewichtung der Rückgabewerte ist daher nicht direkt möglich [96].



Abbildung 2.9: Beispiel zur booleschen Suchanfrage. Bei einer Verknüpfung mit UND müssen alle Ausdrücke gemeinsam vorkommen. Bei der Suche nach Ausdruck A UND B wird Dokument 3 gefunden. Bei einer ODER-Verknüpfung müssen die Ausdrücke nicht gemeinsam vorkommen. A ODER B liefert daher die Dokumente 1, 2 und 3 zurück. Bei der Verknüpfung NICHT B wird das Dokument 1 zurückgeliefert.

Bei natürlichsprachlichen Suchanfragen (engl. *natural language query*) werden ganze Sätze als Suchanfrage eingegeben. Eine mögliche Suchanfrage kann zum Beispiel lauten: "In welchen Dokumenten ist der Blutdruck erhöht?"

Bei einer Thesaurus-basierten Suchanfrage kann der Benutzer aus einer von der Suchmaschine zur Verfügung gestellten Liste aus vorgegebenen Begriffen eine Suchanfrage zusammenstellen.

Begriff-Suchanfragen (engl. *term search*) sind bei der Suche im Internet die am weitesten verbreitete Methode. Der Benutzer gibt im Gegensatz zur natürlichsprachlichen Suchanfrage nur einzelne Begriffe ein, die bei manchen Suchmaschinen auch mit UND und ODER ähnlich einer booleschen Suche kombiniert werden können.

Neben der Art der Suchanfrage kann die Motivation einer Suche als Unterscheidungskriterium herangezogen werden. In [97] werden drei Gründe für eine Suche im Web angegeben:

- Navigationsorientiert: Suchanfrage, um eine Seite/Ort zu finden. Das Ziel dieser Art von Suchanfragen ist meistens eine konkrete Seite (i.e. die Suchanfrage "ORF" hätte als Ziel die Seite <http://www.orf.at>).
- Informationsorientiert: Suchanfragen, denen ein konkretes Informationsbedürfnis

zugrundeliegt. Die Suchanfrage eines Arztes in einem Shared-EHR-System kann unter dieser Kategorie zusammengefasst werden.

- **Transaktionsorientiert:** Suchanfragen, die dazu dienen, Seiten im Netz zu finden, die als Ausgangspunkt für weitere Interaktionen dienen. Dazu zählt die Suche nach Online-Einkaufsportalen oder Webseiten, die Bilder, Lieder etc. anbieten.

Eine Analyse der Logs der Suchmaschine *Alta-Vista* ergab, dass rund 20% der Suchanfragen im Internet navigationsorientiert, rund 48% informationsorientiert und 30% transaktionsorientiert sind [97]. Die Zahlen stammen zwar aus dem Jahr 2002, können jedoch als Anhaltspunkt dienen.

Aufgrund der großen Anzahl an unstrukturierter Information im Web konzentrierten sich die ersten Suchmaschinen auf die Metainformation im Kopfbereich von HTML-Dokumenten. Metainformationen sind Daten, die die Daten selber beschreiben. Bei HTML-Dokumenten befinden sich diese Informationen im Kopfbereich und beinhalten unter anderem den Titel, Autor oder den Inhalt der Datei. Im Falle von Dokumenten können zusätzliche Metadaten wie das Erstellungsdatum, der Dokumententyp, etc. angegeben werden. Diese metadatenbasierte Suche muss von einer Metasuchmaschine [98] unterschieden werden, die dazu dient, Suchanfragen an andere Suchanbieter aufzuteilen und die aggregierten Ergebnisse dem Benutzer zurückzuliefern.

Im Gegensatz zur metadatenbasierten Suche, bei der ganze Dokumente zurückgeliefert werden, können bei der inhaltsbasierten Suche zusätzlich nur die relevanten Teile der Dokumente zurückgeliefert werden. Dies ist vergleichbar mit Suchanfragen auf hochstrukturierte Daten in relationalen Datenbanken. Die Suchanfragen werden in eigenen Abfragesprachen wie z.B. mittels SQL abgesetzt und werden im Falle von SQL in einer tabellarischen Form zurückgeliefert. Die Rückgabetable muss dabei nicht dem ursprünglichen Eingabeformat (z.B. Formular) entsprechen, sondern beinhaltet nur die in der Suchanfrage angegebenen Inhalte.

## **2.4.2 Suche in XML-Dokumenten**

Da EHR-Dokumente in IHE-XDS als XML-Dokumente abgelegt werden, konzentriert sich die vorliegende Arbeit im Folgenden auf Methoden der Informationsbeschaffung von Inhalten in

XML-Dokumenten. Dieses Forschungsgebiet gibt es seit den 1990er Jahren und wie im klassischen IR werden fünf Kategorien von Problemstellungen unterteilt [76]:

- Abfragen (engl. *querying*)
- Indexieren (engl. *indexing*)
- Ranking (engl. *ranking*)
- Präsentation der Ergebnisse (engl. *presenting*)
- Evaluierung (engl. *evaluating*)

Bei der Suche in unstrukturierten Daten können nur Wörter oder Sätze verwendet werden, um nach den gewünschten Inhalten zu suchen. Bei strukturierten Daten kann zusätzlich noch die Struktur in der Suchanfrage verwendet werden. Mithilfe der Struktur können die Orte, in denen die gewünschten Suchwörter vorkommen, festgelegt werden, um ein noch genaueres Suchergebnis zu erhalten. In [99] werden Abfragen, die nur den Inhalt betreffen (engl. *content-only*), und Abfragen, die zusätzlich die Struktur berücksichtigen (engl. *content-and-structure*), unterschieden. Es werden auch Probleme bei der Berücksichtigung der Struktur aufgezeigt. So wird die Struktur eher als Hinweis auf den Suchort verwendet und nicht als verpflichtende Vorgabe. Auch wird durch das Verwenden der Struktur die Suche nicht automatisch besser, da durch semantische Fehler in der Suchanfrage relevante Informationen nicht gefunden werden. Auch ist ein genaues Wissen der Struktur der zu durchsuchenden Dokumente notwendig, um sinnvolle Anfragen zu erzeugen.

Bei der Indexierung werden für gewisse Suchwörter die resultierenden Suchergebnisse in einem Index abgelegt. Bei der Suche in unstrukturierten Dokumenten wird als Ergebnis ein Link auf das Ergebnisdokument im Index hinterlegt, wodurch nicht das ganze Dokument im Index gespeichert werden muss und die Größe des Index gering gehalten werden kann. Bei der Suche in strukturierten Dokumenten, bei denen nicht nur ganze Dokumente als Suchergebnis zurück geliefert werden, müssten alle Kombinationen an Unterbäumen indexiert werden. Dies würde bedeuten, dass das ganze Ergebnisdokument auch im Index abgelegt wird.

Abhängig vom Suchgebiet kann eine unterschiedliche Granularität der Suchergebnisse notwendig sein, um Informationsüberflutung zu vermeiden, den Kontext der Information

jedoch nicht zu verlieren. Das sogenannte *focused retrieval* [100] beschreibt eine Methode, nicht ganze Dokumente, sondern nur die relevanten Teile von Dokumenten als Suchergebnis zurückzuliefern. In [101] wird das System *XRANK* vorgestellt, das im Gegensatz zur HTML-Suche nicht ganze Dokumente, sondern auch Teile eines XML-Dokuments als Resultat zurückliefern kann.

Um Wissenschaftlern die Möglichkeit zu bieten, ihre entwickelten XML-Suchalgorithmen zu vergleichen, wurde 2002 die *INitiative for the Evaluation of XML-Retrieval* (INEX) gegründet [94]. In INEX werden standardisierte Datensätze zur Verfügung gestellt, um die unterschiedlichen Ansätze miteinander vergleichen zu können. Die Datensätze werden von der INEX nur zur Verfügung gestellt und beinhalten zum Beispiel einen *XML-Dump* der Daten aus Wikipedia. Evaluationskriterien können dabei berücksichtigen, ob die zurückgelieferten Ergebnisse alle relevanten Informationen beinhalten und ob diese nicht in einem anderen Abschnitt auch behandelt wurden. Ein kleiner Absatz mit relevanten Informationen wird in [99] als *erschöpfend* (engl. *exhaustive*) bezeichnet und *spezifisch* (engl. *specific*), wenn das Ergebnis nur an dieser einen Stelle gefunden wurde. Wird der ganze Artikel, der den relevanten Absatz beinhaltet, zurückgeliefert, ist das Ergebnis immer noch *erschöpfend*, aber nicht *spezifisch*.

### 2.4.3 Suche im Semantic Web

Das *Resource Description Framework* (RDF) ist ein Datenmodell [102] zur Darstellung von Beziehungen zwischen *Ressourcen*. Die Beziehungen werden dabei als Statements der Form Subjekt (engl. *subject*), Prädikat (engl. *predicate*) und Objekt (engl. *object*) dargestellt [103]. In weiterer Folge werden diese kurz als *RDF-Triple* bezeichnet. Mehrere *RDF-Triple* bilden dabei einen gerichteten Graphen, dessen Richtung durch das Prädikat vorgegeben ist und in RDF-Dokumenten abgelegt ist. Ein Vorteil der gerichteten Graphen ist, dass sie leicht vereint werden können. Um Ressourcen eindeutig zu bezeichnen und ein Vereinen von unterschiedlichen RDF-Dokumenten zu ermöglichen, wird jede Ressource mit einem *Uniform Resource Identifier* (URI) eindeutig beschrieben. URI (engl. für "einheitlicher Bezeichner für Ressourcen") wurden entwickelt, um Ressourcen im Web eine eindeutige Bezeichnung zu geben. Es können aber auch Objekte, die nicht im Web sind, wie zum Beispiel Archetypen oder die Klassen eines Referenzmodells, damit eindeutig referenziert werden. Um sicherzustellen, dass eine URI wirklich eindeutig ist, sind URIs von einer Webseite abzuleiten, die unter der

eigenen Kontrolle ist (z.B. der Webserver der Forschungsgruppe MIAS <https://mias.meduniwien.ac.at/XYZ>). In sogenannten *Triple Stores* werden die *RDF-Triple* für Auswertungen zur Verfügung gestellt. *RDF-Triple* können unterschiedlich serialisiert werden. XML wird im Rahmen dieser Arbeit zur Darstellung aller EHR-Dokumente verwendet.

Mittels *RDF-Schema* (RDFS) wird das Vokabular, das in RDF verwendet wird, beschrieben. Neben dem Vokabular kann mit RDFS festgelegt werden, welche Eigenschaften auf welche Objekte zutreffen. Zusätzlich sind Einschränkungen der gültigen Ausprägungen von Objekten und der Beziehungen zwischen Objekten möglich. Die *Web Ontology Language* (OWL) baut auf RDF bzw. RDFS auf, die Eigenschaften von Klassen können jedoch noch detaillierter beschrieben werden.

Die *SPARQL Protocol And RDF Query Language* (SPARQL) ist eine Abfragesprache für *RDF-Triple*. Eine zusätzliche Abfragesprache zu den XML-Abfragesprachen ist nötig, da idente Aussagen in RDF unterschiedlich dargestellt werden können. Dies hätte zur Folge, dass unterschiedliche XPATH-Anweisungen für dieselbe Abfrage erzeugt werden müssten [102]. SPARQL basiert auf der Suche von Mustern in Graphen.

In [104] werden die Technologien des Semantic Web als Lösungsansätze für die Suche in Shared-EHR-Systemen und Shared-EHRs gesehen. Eine Suche mit Semantic-Web-Technologien erlaubt den zu suchenden Text für Computer auf einer semantischen Ebene verständlich zu machen. Agenten verstehen die medizinischen Inhalte und können neues Wissen generieren. Auch die Beantwortung von Fragestellungen mit Information aus mehreren Dokumenten könnte dadurch ermöglicht werden.

Um die existierenden Technologien des *Semantischen Webs* anzuwenden, müssen jedoch alle Inhalte mit *RDF-Triple* dokumentiert werden. Diese Dokumentation ist sehr zeitaufwendig und in der Routedokumentation nicht vorstellbar. Aktuelle Standards wie CDA oder ISO/EN 13606 setzen nicht auf RDF. In [105] wird ein Überblick gegeben, wie *RDF-Triple* und die CDA gemeinsam verwendet werden können und welche Vorteile RDF für Shared-EHRs haben kann.

#### **2.4.4 Suche in Shared-EHR-Systemen**

Neben der in IHE-XDS angebotenen metadatenbasierten Suche in Shared-EHR-Systemen, finden sich in der Literatur noch weitere Ansätze zur Informationssuche in Dokumenten. In diesem Abschnitt werden existierende Suchen, die speziell für Shared-EHR-Systeme konzipiert

wurden, gezeigt. Da die Suche mithilfe von Archetypen und IHE-XDS analysiert wird, werden auch Suchen basierend auf Archetypen und andere IHE-Profile, die die Suche von medizinischen Inhalten abdecken, besprochen.

Die Suche in Shared-EHR-Systemen, in denen Shared-EHRs zur Verfügung gestellt werden, unterscheidet sich stark von der Suche im Web, da die Menge an zu durchsuchenden Dokumenten um viele Größenordnungen kleiner ist. Im Web müssen alle existierenden Webseiten durchsucht werden, im Anwendungsfall von Shared-EHR-Systemen handelt es sich um die Dokumente einer einzelnen Person, die durchsucht werden. Ein weiterer wichtiger Aspekt bei der Suche ist das Ranking der Inhalte. In der Suche von IHE-XDS werden die Dokumente chronologisch geordnet, da aktuellere Daten in der Medizin in der Regel eine höhere Relevanz besitzen. Das Vermeiden von *Spam* und die hohe Heterogenität der Inhalte bei der Suche im Internet spielen bei der Suche in Shared-EHR-Systemen keine so große Rolle. Shared-EHRs basieren auf einem akkordierten Informationsmodell. Die Struktur, die dadurch vorgegeben ist, kann bei der Suche berücksichtigt werden. Der Großteil der Dokumente im Internet ist im Gegensatz dazu unstrukturiert.

Bei der Suche in Shared-EHR-Systemen ist es wichtig, die Informationsbedürfnisse von Ärzten zu verstehen, um Informationsüberflutung zu vermeiden. In [106] werden Suchanfragen in einem EHR-System analysiert. Über eine Logfile-Analyse wurden über einen Zeitraum von 6 Monaten über 2000 Suchanfragen gesammelt. Die Analyse ergab, dass über 85% der Suchanfragen informationsorientiert waren und vorwiegend Laborergebnisse sowie spezielle Diagnosen gesucht werden. Dies steht im Gegensatz zu den 48% informationsorientierter Suchen, die in [97] für die Suche im Internet ermittelt wurden. Abhängig vom Einsatzgebiet der Suche kann der Fokus der Suche (i.e. informationsorientiert, navigationsorientiert, transaktionsorientiert) sich ändern und muss bei der Suche berücksichtigt werden.

#### **2.4.4.1 Archetypen-unabhängige Suche**

*MorphoSaurus* [107] ist eine Dokumentensuchmaschine, die in ein EHR-System integriert wird. Sie erlaubt es, eine Volltextsuche in Arztbriefen und anderen relevanten medizinischen Dokumenten auszuführen. Es werden sowohl die Suchanfragen als auch die Dokumente mithilfe einer morphologischen Analyse und semantischer Indizierung aufbereitet. *MorphoSaurus* beinhaltet rund 100.000 lexikalische Einträge und wurde mit einem 30.000

Dokumente umfassenden Dokumentensatz getestet.

Im Rahmen des *RAVEL*-Projekts [108] in Frankreich werden drei Aspekte bei der Suche in EHR-Dokumenten erforscht: die semantische Indizierung von EHR-Daten, die Suche selber sowie das Visualisieren der Suchergebnisse. Die relevanten vollstrukturierten und unstrukturierten Dokumente werden in einem zentralen Repository gespeichert. Der Suchindex wird mit Methoden der Computerlinguistik in Kombination mit medizinischen Ontologien aufgebaut. Es sollen natürlichsprachliche Suchanfragen unterstützt werden. Es werden weder Angaben zum Standard der verwendeten EHR-Dokumente (z.B. CDA, ISO/EN 13606) noch zur Architektur, zur Speicherung und zur Suche der Dokumente (z.B. IHE-XDS) gemacht.

Im *mCHAS*-Projekt [109] wurde ein Prototyp eines EHR-Systems entwickelt, der im Guangdong Krankenhaus für Traditionelle Chinesische Medizin eingesetzt wird. Die medizinischen Inhalte sind dabei in einer IHE-XDS-konformen Architektur als CDA-Dokumente abgelegt. Für das Erstellen des Index werden die *Code-Elemente*, die in allen *Act-Klassen* über das RIM zugelassen sind (z.B. *Observation*, etc.), extrahiert und in ein *RDF-Triple* umgewandelt. Im Index werden diese *RDF-Triple* mit denen der Referenzterminologie kombiniert. Über eine webbasierte Benutzeroberfläche können Suchanfragen an die semantische Suchmaschine abgesetzt werden. Diese kombiniert die metadatenbasierte Suche von IHE-XDS und die *RDF-Triple* und liefert eine Liste relevanter Dokumente zurück. Das *Document Repository* wird durch diese Methode mit zusätzlichen Informationen über Inhalte ergänzt, wodurch eine inhaltliche Suche in IHE-XDS umsetzbar ist.

Eine Implementierung vom *Peking University People's Hospital* (PKUPH) wird in [110] präsentiert. Um die Komplexität von CDA zu verringern, wird ein Ansatz ähnlich der *Green CDA* gewählt. Einfachere XML-Strukturen werden durch eine Transformation in valide CDA-Dokumente umgewandelt. Diese Implementierung baut ebenfalls auf IHE-XDS auf, bietet aber neben dem *Standard-Document Repository* ein zusätzliches *EHR Data Repository* an, um eine inhaltsbasierte Suche zu ermöglichen. Das *EHR Data Repository* baut dabei auf eine hybride XML-Datenbank auf; das bedeutet, dass XML direkt in relationalen Strukturen gespeichert werden kann. Diese Vorgehensweise erlaubt die Vorteile von relationalen Datenbanken und XML-Datenbanken zu kombinieren und schnell auf relevante Ausschnitte eines XML-Dokuments zuzugreifen.

Ebenfalls auf IHE-XDS aufbauend wird in [111] ein Ansatz präsentiert, um zusätzlich zu den

Metadaten, die das Dokument beschreiben, Metadaten über den Inhalt von verschlüsselten EHR-Dokumenten zugänglich zu machen. Mittels vollautomatischer Erkennung von Textbedeutungen und der *Unified Medical Language System (UMLS)* Metathesaurus [46] werden alle gespeicherten Dokumente analysiert und zusätzlich zu der *Document Registry* ein *Meta-information Index* angelegt. Dabei werden nur die medizinischen Konzepte und nicht die Ausprägungen selber im Index gespeichert. Durch das Kombinieren der *Document Registry* und des *Meta-information Index* kann bei Suchanfragen die Relevanz der zurückgelieferten Dokumente erhöht werden.

Das System *XOntoRank* [112] ermöglicht eine semantische Suche in EHR-Dokumenten. CDA-Dokumente wurden aus einem bestehenden System automatisch generiert und indiziert. Die semantische Suche erweitert die Suchanfrage mithilfe von medizinischen Ontologien wie Snomed CT, um auch ähnliche Konzepte zum Suchausdruck zu finden. Dies erlaubt es, zusätzliche Suchergebnisse zurückzuliefern. Die Auswahl der zurückgelieferten Werte sowie das Ranking der gefundenen Ergebnisse hängen dabei von der verwendeten Ontologie ab.

In Tabelle 2.1 werden die vorgestellten Suchen in Bezug auf die Art der Suche, die Art des verwendeten Index und den zugrundeliegenden Daten gegenübergestellt. Außer [111], setzen alle Ansätze bei der inhaltsbasierten Suche auf einen zentralen Index, in dem die Dokumentinhalte indiziert sind.

#### **2.4.4.2 Archetypen-spezifische Suche**

In [113] wird die *EHR Query Language (EQL)* die später in *Archetype Query Language (AQL)* umbenannt wurde, vorgestellt. Die AQL ist auf die Bedürfnisse des Zwei-Modell-Ansatzes angepasst. Abfragen können ähnlich zu SQL-Suchanfragen formuliert werden, wobei der ADL-Pfad-Mechanismus zum Definieren der Auswahlkriterien und Rückgabewerte verwendet wird. AQL-Abfragen können mithilfe des *Query Builder* [114] erzeugt und von einem proprietären EHR-Server, in dem die archetypbasierten Dokumente gespeichert sind, ausgewertet werden.

Die AQL erlaubt es, unabhängig vom Format, in dem das Shared-EHR persistiert wurde, Suchanfragen zu formulieren. OpenEHR- sowie ISO/EN-13606-EHR-Extrakte können zwar wie CDA-Dokumente als XML-Dokumente gespeichert werden, können aber ebenso als Objekte in einer Programmiersprache (z.B. Java-Objekte) persistiert werden.

<b>Name (Referenz)</b>	<b>Art der Suche</b>	<b>Art des Index</b>	<b>EHR-Dokumente</b>
<i>XDS</i> [16]	metadatenbasierte Suche	zentraler Index mit Metadaten	CDA (Level 1-3)
<i>MorphoSaurus</i> [107]	inhaltsbasierte Suche	zentraler Index mit Dokumentinhalten	proprietäre Arztbriefe (Level 1)
<i>RAVEL</i> [108]	inhaltsbasierte Suche	zentraler Index mit Dokumentinhalten	nicht näher spezifiziert, Level 2 und 3
<i>mCHAS</i> [109]	inhaltsbasierte Suche	zentraler Index mit Dokumentinhalten	CDA (Level 3)
<i>PKUPH</i> [110]	inhaltsbasierte Suche	zentraler Index mit Dokumentinhalten	CDA (Level 3)
<i>Verschlüsselte EHR-Daten</i> [111]	inhaltsbasierte Suche	zentraler Index mit dokumentspezifischen Metainformationen	CDA (Level 2)
<i>XOntoRank</i> [112]	inhaltsbasierte Suche	zentraler Index mit Dokumentinhalten	CDA (Level 2 und 3)

Tabelle 2.1: Gegenüberstellung der Archetyp-unabhängigen Suchen

XQuery-Abfragen können nur auf XML-Dokumente abgesetzt werden. Im Gegensatz zu AQL ist XQuery ein anerkannter Standard, für den in vielen Programmiersprachen Tools zur Verfügung gestellt werden. Eine genaue Gegenüberstellung von AQL und XQuery kann in [115] gefunden werden.

Der in [116] vorgestellte Ansatz beruht auf *REST*-Technologien (Webservices mit zusätzlichen Eigenschaften), um Bookmarks auf EHR-Inhalte zu erstellen. Dabei kann für jeweils einen einzelnen Knoten ein Bookmark erstellt werden. Die Bookmarks basieren dabei wie die AQL auf dem ADL-Pfad-Mechanismus, der erweitert wurde, um zusätzlich Versionsinformationen im Shared-EHR zu berücksichtigen. Die daraus generierten URLs dienen dazu, über die *REST*-Technologie den gewünschten Inhalt von einem Server zu laden. Die URL kann dabei als einfache Suchanfrage gesehen werden, die jedoch keine logischen Operatoren oder andere Mechanismen unterstützt.

In [61] wird das System *LinkEHR* vorgestellt. *LinkEHR* bietet neben der Möglichkeit, Archetypen zu erstellen, in der kommerziellen Version zusätzlich eine Mappingfunktion an, um existierende Daten u.a. als ISO/EN-13606-EHR-Extrakte zu exportieren. Das Mapping basiert

dabei auf Archetypen. Ob und wie eine Suche mit *LinkEHR* möglich ist, wird nicht erwähnt.

Im *ByMedConnect* System [35, 36] werden ISO/EN-13606-EHR-Extrakte zwischen Arztpraxen in Bayern ausgetauscht. Es handelt sich dabei im Gegensatz zu IHE-XDS um eine gerichtete *Peer-to-Peer-Kommunikation* mit einem zentralen Kommunikationsserver. Wie schon beim System *LinkEHR* liegt der Fokus von diesem System im standard-konformen Austausch von EHR-Dokumenten. Da nur einzelne EHR-Dokumente über das System ausgetauscht werden, gibt es keine Möglichkeit, das gesamte EHR eines Patienten zu durchsuchen.

#### **2.4.4.3 IHE-Profil für inhaltsbasierte Suchen**

Neben IHE-XDS gibt es von IHE noch weitere Profile, die sich mit der Suche in Dokumenten auseinandersetzen.

Das IHE-Profil *Query for Existing Data* (QED) [117] erlaubt es unter anderem, Abfragen auf die Inhalte der *Continuity of Care Documents* (CCD-Dokumente) [118] zu machen. Dazu gehören Vitalparameter, Probleme, Medikationen, Impfungen, Diagnose, Behandlungen und eine Auflistung der Besuche. Suchanfragen werden mittels eines *transmission wrapper* versendet, der unter anderem Informationen über den Empfänger und den Sender sowie den eigentlichen Inhalt mittels *Control Act Wrapper* beinhaltet. Der *Control Act Wrapper* beinhaltet unter anderem Informationen über die Personen, die an der Transaktion beteiligt sind, und die eigentliche Suchanfrage in der *Parameter List*. In der *Parameter List* können folgende Informationen eingeschränkt werden: Daten zum Patienten (i.e. ID, Geburtsdatum, Geschlecht, Name), Zeitbereiche (i.e. wann das Dokument erzeugt wurde und wann dokumentierte medizinische Handlungen stattgefunden haben), Anzahl der Rückgabewerte sowie ein Code, der angibt, welche Teile der CCD gewünscht werden. Es gibt z.B. Codes für “alle Allergien”, “alle Laborbefunde”, aber auch einzelne Vitalparameter wie zum Beispiel der Puls können mittels Code abgefragt werden. Im QED-Profil werden die Zeitbereiche wie in Abbildung 2.10 gezeigt behandelt. Dokumente B und D, die teilweise in den Zeitbereich fallen, sowie Dokument C werden angezeigt. Dokumente A und E werden jedoch nicht zurückgeliefert. Im Rahmen der vorliegenden Arbeit wurde dieser Ansatz ebenfalls verwendet.

Die QED-Spezifikation befindet sich seit 2008 im Entwurfs-Stadium. In [119] wird die QED-Spezifikation analysiert und die große Bedeutung des Kontexts der medizinischen

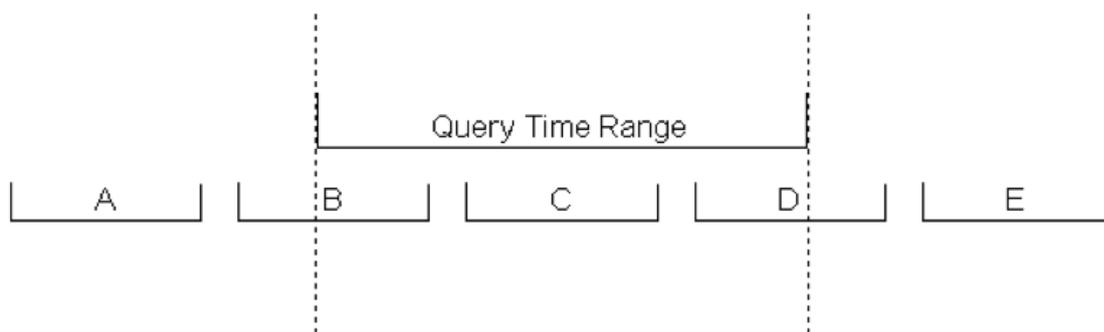


Abbildung 2.10: Umgang mit Zeitintervallen im QED-Profil (aus [117]).

Dokumentation hervorgehoben, der bei QED-Suchanfragen verloren geht. Es ist zum Beispiel nicht mehr ersichtlich, aus welchen Dokumenttypen ein Ergebnis einer Suchanfrage generiert wird. Weiters können nur Dokumente von einer *Document Source* in dieser Weise abgefragt werden, eine übergreifende Suche über mehrere *Document Sources* ist nicht möglich. Ein ähnlicher Ansatz wird im IHE-Profil *Care Management* (CM) [120] und im *Retrieve Information for Display* (RID) Profil angewendet.

IHE-XDS unterstützt nur Anfragen, die genau einen Patienten betreffen. Für epidemiologische Aufgabenstellungen kann es aber auch nötig sein, nach Patienten mit einer bestimmten Krankheit zu suchen. Für diesen Zweck wurde das IHE-Profil *Multi-Patient Queries* (MPQ) [121] entwickelt. Es befindet sich momentan im Entwurfs-Stadium (engl. *Trial Implementation*). Es können aggregierte anonymisierte Daten sowie in einer detaillierten Anfrage auch nicht anonymisierte Daten abgefragt werden. Die Suchanfrage wird wie bei IHE-XDS mit der *Stored Query transaction* (Transaktion *ITI-18*) durchgeführt.

Im IHE-Profil *On-Demand Documents* [122] werden zusätzlich zu den statischen Dokumenten, die in IHE-XDS versendet werden, noch dynamische Dokumente angeboten, deren Inhalt sich von Abfrage zu Abfrage verändern kann. Die *On-Demand-Dokumente* können aus einer *Document Source* oder aus Dokumenten von mehreren *Document Sources* zusammengesetzt werden. *On-Demand-Dokumente* können ähnlich wie vorgefertigte Suchanfragen aufgebaut sein.

### 2.4.5 Aufbereitung der Suchergebnisse

Neben den Ansätzen zur Suche selber, ist auch die Darstellung der zurückgelieferten Suchergebnisse ein wichtiges Forschungsgebiet. Die Ergebnisse, die im ersten Schritt gefunden werden, müssen in weiterer Folge in einer übersichtlichen Form dargestellt werden. Bei unstrukturierten Dokumenten werden die gefundenen Dokumente aufgelistet, strukturierte Dokumente ermöglichen es durch das *focused retrieval*, die relevanten gefundenen Ausschnitte speziell aufzubereiten und so eine angepasste Darstellung zurückzuliefern.

In [123] wird ein Überblick über die Visualisierung medizinischer Daten, die direkt bei der Patientenbehandlung eingesetzt wird, gegeben. Der Fokus liegt dabei in der Visualisierung der zeitlichen Abfolgen bei der Patientenbehandlung. Unter anderem wird auch das Framework von [124] zum Visualisieren eines EHRs eines Patienten über eine Zeitleiste (engl. *time line*) analysiert. In diesem Ansatz werden die medizinischen Informationen aus unterschiedlichen Systemen als XML-Instanzen gesammelt und in ein proprietäres, problemorientiertes Format gebracht. Über die Zeitleiste bekommen Ärzte einen Überblick über die medizinischen Daten des jeweiligen Patienten.

Speziell auf CDA und semantische Web-Technologien ist der Ansatz in [125] aufgebaut. CDA-Dokumente werden mit der Ontologie in [126] in *OWL*-Instanzen umgewandelt. Durch diese Transformation können die Technologien des Semantischen Web wie etwa SPARQL zur Suche in den Daten angewendet werden. Die SPARQL-Ergebnisse werden in standardisierte XML-Dokumente umgewandelt, die direkt mit im Internet verfügbaren Java-Script-Bibliotheken dargestellt werden können. So wird der Ansatz mit der vom *Massachusetts Institute of Technology* (MIT) entwickelten *Timeline* [127] und mit der JavaScript-Diagrammbibliothek *amCharts* [128] getestet. Dieser Ansatz hat den Vorteil, dass existierende Tools zur Visualisierung verwendet werden können.

Das Forschungsgebiet *Aggregated Search* [129] beschäftigt sich damit, die Ergebnisse unterschiedlicher spezialisierter Suchen in einem Ergebnis zusammenzufassen. Die Ergebnisse können dabei Zeitleisten, Diagramme, tabellarische Darstellungen sowie Textauschnitte beinhalten. Dieser Ansatz wird mittlerweile auch von Google bei der Websuche angewendet, indem Landkarten zu den Suchergebnissen, Blogbeiträge, aktuelle Nachrichten, Videos und Bilder basierend auf den eingegebenen Suchwörtern zusammen angezeigt werden.

Durch das Extrahieren einzelner Teile bei der Visualisierung von EHR-Suchergebnissen ist es schwer, den Kontext der dargestellten Ergebnisse eindeutig zu identifizieren. Durch das zusätzliche Darstellen eines Inhaltsverzeichnisses kann der Kontext der Ergebnisse erhalten bleiben. In [130] wird beschrieben, was bei der automatischen Generierung eines Inhaltsverzeichnisses berücksichtigt werden muss. So sollte zum Beispiel das Inhaltsverzeichnis auch bei langen Dokumenten nicht länger sein als bei kurzen Dokumenten und die Elemente des Inhaltsverzeichnisses die Suchanfrage selber berücksichtigen.

In [131] wird ein Ansatz für die Visualisierung von archetypbasierten EHR-Dokumenten beschrieben. Dieser Ansatz ist nicht speziell für die Visualisierung von Suchergebnissen konzipiert, die in XML beschriebene Visualisierung bestimmter Archetyp-Knoten kann jedoch als Basis für die Visualisierung der Suchergebnisse herangezogen werden. Das Wissen über die Visualisierung ist dabei vom medizinischen Wissen in Archetypen getrennt und geht auf die Anforderungen der unterschiedlichen Endgeräte wie TV, Smartphone oder PC an eine Visualisierung ein.



## Methoden

Die im Folgenden vorgestellten Methoden nutzen die Eigenschaften von EHR-Dokumenten, die nach dem Zwei-Modell-Ansatz aufgebaut sind, und verwenden Archetypen als zentrale Komponenten. Der Hauptfokus liegt auf der Methode zur Umsetzung einer inhaltsbasierten Suche in Shared-EHR-Systemen mit zentraler Metadatenkomponente. Im Gegensatz zu den in Kapitel 2.4.4 vorgestellten Suchen, die bis auf einen Ansatz auf einem zentralen Index mit Dokumentinhalten aufbauen, wird kein zentraler Index mit detaillierten Informationen zu Dokumentinhalten aufgebaut, da dies dem Ansatz der dezentralen Datenhaltung widerspricht und ein zusätzliches Datenschutzrisiko bedeutet. Für die Erzeugung der im Zwei-Modell-Ansatz benötigten Beschreibungen des Wissens wird eine Methode aufbauend auf [132] präsentiert. Aufbauend auf der *Plug-and-Play-Integration von Archetypen* [52] wird eine Methode präsentiert, um die fein-granularen EHR-Dokumente zu erstellen. Weiters wird eine Methode zum semantischen Validieren der so erzeugten EHR-Dokumente mittels XML-Schema vorgestellt.

### **3.1 Inhaltsbasierte Suche in Shared-EHR-Systemen mit zentraler Metadatenkomponente**

In diesem Abschnitt wird eine Methode zur inhaltsbasierten Suche in Shared-EHR-Systemen mit zentraler Metadatenkomponente aufbauend auf den Konzepten des Zwei-Modell-Ansatzes präsentiert. Bei bestehenden Ansätzen zur inhaltsbasierten Suche in Shared-EHR-Systemen

wird ein zentraler Index angelegt, in dem die Suchwörter indiziert sind [108, 109, 110, 112]. In diesen zentralen Indices werden zusätzliche Informationen zu den Dokumentinhalten zentral abgelegt. Das erlaubt es, ohne die Dokumente einzeln zu durchsuchen, schnell relevante Information als Antwort auf eine Suchanfrage zurückzuliefern. Ein wesentlicher Nachteil dieses Ansatzes besteht jedoch darin, dass durch den zentralen Index die Strategie der dezentralen Datenhaltung von Shared-EHR-Systemen mit zentraler Metadatenkomponente unterlaufen wird und der Vorteil dieser Architektur verloren geht, Datenmissbrauch durch die Absenz zentral gespeicherter Dokumentinhalte zu vermeiden.

Im Gegensatz dazu verwendet der in der vorliegenden Arbeit vorgeschlagene Ansatz nur Wissen über die Dokumentstruktur, um einen Index der Dokumenttypen zu erstellen. Über die Beschreibung von Dokumenttypen kann der Aufbau von EHR-Dokumenten bestimmt werden. Ein Dokumenttyp kann zum Beispiel ein Arztbrief (engl. *Discharge Summary*) oder ein Laborbefund (engl. *Laboratory Report*) sein. Ein vollstrukturiertes EHR-Dokument ist von einem gewissen Dokumenttyp, der durch einen Wurzel-Archetypen und beliebig viele Unter-Archetypen, die über *Slots* eingebunden sind, beschrieben ist. Dabei können auch Platzhalter (engl. *Wildcards*) angegeben werden, um einen beliebigen Unterbaum zu erlauben. Ein Index über die Dokumenttypen bereitet das Wissen über den Aufbau der Dokumenttypen so auf, dass spezifische Inhalte gezielt in den jeweiligen Dokumenten gefunden werden können. Im Gegensatz zu einem Index der Dokumentinhalte, der Dokumente oder Teile eines Dokumentes eines Patienten zentral abspeichert, werden beim Index der Dokumenttypen keine medizinischen Inhalte eines Patienten zentral gespeichert.

In Abbildung 2.4 auf Seite 35 wird die metadatenbasierte Suche in einer dezentralen Datenhaltung mit Metadatenkomponente schematisch gezeigt. Der in Abbildung 3.1 präsentierte Ansatz für die inhaltsbasierte Suche erweitert die metadatenbasierte Suche um die durch strichlierte Linien gekennzeichneten Komponenten. Es ist ersichtlich, dass alle für die metadatenbasierte Suche verwendeten Komponenten und Schnittstellen unverändert für die inhaltsbasierte Suche verwendet werden.

Für eine inhaltsbasierte Suche werden im ersten Schritt wie bei der metadatenbasierten Suche die Dokumente mit ihren Metadaten in der zentralen Metadatenkomponente registriert. Im zweiten Schritt werden inhaltsbasierte Suchanfragen aus Suchbegriffen, die von der Wissensbasis geladen werden, erzeugt. Im Gegensatz zu der herkömmlichen

metadatenbasierten Suche, die von den GDAs direkt an die Metadatenkomponente gesendet wird, werden im dritten Schritt inhaltsbasierte Suchanfragen an die Wissensbasis gesendet. Mithilfe der Wissensbasis wird die inhaltsbasierte Suchanfrage in eine metadatenbasierte Suche und eine auf die EHR-Dokumente angepasste XQuery-Suchanfrage umgewandelt und an den GDA zurückgesendet. Im vierten Schritt wird die so erzeugte metadatenbasierte Suche an die Metadatenkomponente gesendet und die Metadaten inklusive Speicherort zurückgeliefert. Im fünften Schritt werden die EHR-Dokumente von den jeweiligen GDAs heruntergeladen. Im sechsten Schritt wird die in Schritt drei erzeugte XQuery-Suchanfrage auf die EHR-Dokumente angewendet und das Ergebnis dem GDA präsentiert.

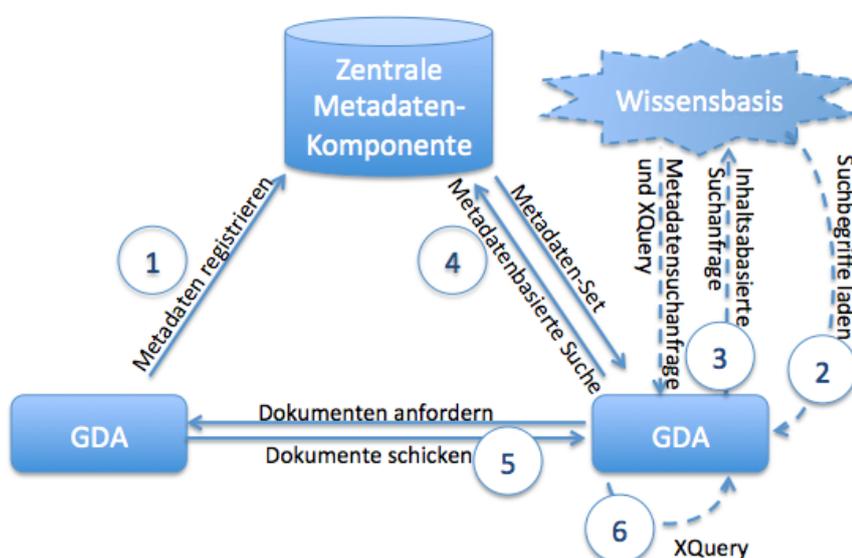


Abbildung 3.1: Übersicht der dezentralen Datenhaltung mit zentraler Metadatenkomponente mit zusätzlicher inhaltsbasierter Suche.

Für die Schritte 1, 4 und 5 in Abbildung 3.1 kann auf existierende Implementierungen in IHE-XDS zurückgegriffen werden. Die an diesen Schritten beteiligten Komponenten (i.e. *Document Source*, *Document Registry*, *Document Repository*) werden daher nicht detailliert beschrieben. Der Fokus liegt im Folgenden auf den Anforderungen an die Wissensbasis, dem Erzeugen der inhaltsbasierten Suchanfrage, dem Erzeugen der metadatenbasierten Suchanfrage, dem Erzeugen der XQuery-Suchanfrage und dem Aufbereiten der Suchergebnisse.

Die vorgestellte Methode ist allgemein für EHR-Standards, die auf dem Zwei-Modell-Ansatz

aufbauen, anwendbar und kann in existierende Shared-EHR-Systeme mit zentraler Metadatenkomponente eingebunden werden. Einzelne GDAs können in ihren Systemen eine inhaltsbasierte Suche anbieten, ohne die Systeme von anderen GDAs, welche keine inhaltsbasierte Suche verwenden, zu beeinflussen. Die vorgestellte Methode zur inhaltsbasierten Suche in Shared-EHR-Systemen geht von folgenden Voraussetzungen aus:

- Das Shared-EHR-System basiert auf einer dezentralen Datenhaltung mit einer zentralen Metadatenkomponente.
- In der Metadatenkomponente sind neben den allgemeinen Metadaten wie Dokumentdatum, Autor etc. nur wenige domänenspezifische Metadaten wie Dokumenttyp oder Fachgebiet des Arztes eingetragen um Datenmissbrauch zu minimieren.
- Es existiert kein zentraler Index der Dokumentinhalte.
- Die abgelegten Dokumente basieren auf einem aktuellen EHR-Standard, der den Zwei-Modell-Ansatz umsetzt, wie zum Beispiel ISO/EN 13606, HL7 CDA oder openEHR.
- Die Inhalte der ausgetauschten EHR-Dokumente sind zwischen den GDAs harmonisiert.

### **3.1.1 Die Wissensbasis**

Die Wissensbasis ist die zentrale Komponente für die Umsetzung der inhaltsbasierten Suche in Shared-EHR-Systemen mit Metadatenkomponente. Die Wissensbasis baut auf dem Wissen auf, welches im Zwei-Modell-Ansatz mittels computerverarbeitbaren Wissensartefakten wie ISO/EN-13606-Archetypen, openEHR-Archetypen oder HL7-Templates oder als Freitext wie zum Beispiel in den CDA-Implementierungsleitfäden dargestellt wird. Für die Kommunikation mit den an der Suche beteiligten Akteuren muss die Wissensbasis über Schnittstellen nach außen verfügen. Da unterschiedlichste Akteure auf die Wissensbasis zugreifen, wird die Kommunikation mit Web Services umgesetzt. Web Services sind direkt von den Systemen der GDAs aufrufbar und ermöglichen es mit geringem Aufwand, Anfragen an die Wissensbasis zu senden und Antworten zurückzubekommen.

Folgende Aufgaben können mithilfe der Wissensbasis durchgeführt werden:

- Inhaltsbasierte Suchanfragen formulieren

- Inhaltsbasierte Suchanfragen in metadatenbasierte Suchanfragen umwandeln
- Inhaltsbasierte Suchanfragen in XQuery-Suchanfragen umwandeln, die an die jeweiligen EHR-Dokumente und den zugrundeliegenden Standard angepasst sind
- Suchergebnisse basierend auf dem Wissen aufbereiten
- Information zum automatischen Aufbau von Formularen zum Erzeugen von strukturierten Dokumenten bereitstellen
- Information zum Erzeugen der XML-Schema-Skripte zum Validieren der EHR-Dokumente bereitstellen

Ausgehend von den beschriebenen Aufgaben müssen folgende Anforderungen von der Wissensbasis erfüllt werden.

Um Dokumente basierend auf unterschiedlichen Standards, die auf dem Zwei-Modell-Ansatz aufbauen, suchen zu können, ist die Wissensbasis vom Referenzmodell unabhängig umzusetzen. Werden unterschiedliche Dokumenttypen mit unterschiedlichen Standards abgebildet oder kommen semantisch idente Inhalte in unterschiedlichen Dokumenttypen vor, werden die semantisch identen Konzepte über einen eindeutigen Code in der Wissensbasis zusammengefasst. Diese Codes können aus medizinischen Terminologien und Ontologien wie Snomed CT stammen oder aus frei definierten Code-Listen.

Die oben genannten Aufgaben setzen ein detailliertes Wissen über die Struktur der zugrundeliegenden Daten voraus. Im Zwei-Modell-Ansatz kann zwischen zwei Struktur-Ebenen unterschieden werden. Durch das erste Modell, das Referenzmodell, wird eine statische Struktur vorgegeben, die über alle EHR-Dokumente identisch ist. Der hierarchische Aufbau mit den erlaubten Elementen und Attributen ist klar vorgegeben und kann direkt in den aufgebauten Index einfließen. Durch das zweite Modell, das Wissensmodell, wird die Struktur, die vom Referenzmodell vorgegeben wird, noch zusätzlich eingeschränkt. Diese detaillierte hierarchische Beschreibung der Struktur und Inhalte der EHR-Dokumente ermöglicht es, strukturelle Information bis auf die Ebene einzelner medizinischer Konzepte zu bekommen und diese ähnlich einem Index aufbereitet zur Verfügung zu stellen. Die Wissensbasis muss sowohl die Struktur des jeweiligen Referenzmodells als auch die Struktur der Wissensartefakte bereitstellen können.

Das Wissen in der Wissensbasis wird für das Formulieren der inhaltsbasierten Suchanfrage verwendet. Für das Formulieren der Suchanfragen werden die dokumentierten medizinischen Konzepte für das Erzeugen der inhaltsbasierten Suchanfrage von der Wissensbasis zurückgeliefert. In der Wissensbasis sind zu den unterschiedlichen Inhalten in den Dokumenttypen Informationen zum jeweiligen Inhalt hinterlegt. Dies können einfache textuelle Informationen oder auch Codes zu medizinischen Terminologien und Ontologien sein. Um eine vom Referenzmodell unabhängige Suche zu ermöglichen, werden dabei nicht die einzelnen Knoten der Dokumenttypen zurückgeliefert, sondern die Codes der zugrundeliegenden Konzepte.

Die aus dem Wissen der Wissensbasis generierte inhaltsbasierte Suchanfrage kann mithilfe der Wissensbasis in eine metadatenbasierte Suchanfrage umgewandelt werden. Dies ist möglich, da jeder Begriff in der inhaltsbasierten Suchanfrage einem oder mehreren Knoten eines Dokumenttyps zugeordnet werden kann. Ausgehend vom jeweiligen Knoten können die für die metadatenbasierte Suche benötigten Dokumenttypen extrahiert werden. Die Information, welche medizinischen Inhalte mit welchen Dokumenttypen verknüpft sind, kann bei der Optimierung der metadatenbasierten Suche herangezogen werden.

Die Wissensbasis wird verwendet, um die inhaltsbasierte Suchanfrage in eine XQuery-Suchanfrage umzuwandeln. Die XQuery-Suchanfrage ist an die jeweiligen Standards und Dokumenttypen angepasst und extrahiert die gewünschten Inhalte aus den EHR-Dokumenten. Die in Kapitel 2.1 präsentierten EHR-Standards können als XML-Dokumente dargestellt werden und daher auch mit XQuery durchsucht werden.

Um komplexere inhaltsbasierte Suchanfragen, die unterschiedliche Begriffe mit booleschen Operatoren verbinden, in eine XQuery-Suchanfrage umzuwandeln, muss die Wissensbasis auch Funktionen zum Finden gemeinsamer Elternknoten der mit UND verknüpften Begriffe zur Verfügung stellen. Bei UND-Verknüpfungen, die das Vorhandensein aller durch UND verknüpften Begriffe fordern, wird der gemeinsame Elternknoten für die Suche benötigt, um den Kontext der UND-Verknüpfung zu bekommen. Der gefundene Elternknoten wird zurückgeliefert und in weiterer Folge als Ausgangspunkt für die weitere Verarbeitung herangezogen.

Informationen zur Struktur und den Inhalten der Dokumente können bei der Visualisierung der Suchergebnisse verwendet werden. Bei diesem Schritt können geeignete Visualisierungen in

Abhängigkeit von den Eigenschaften der anzuzeigenden Knoten berücksichtigt werden. Dies erlaubt es, numerische Werte als Graphen oder Diagramme darzustellen oder auch pathologische Werte speziell zu kennzeichnen.

Neben den erwähnten Aufgaben, die für die Umsetzung der Suche benötigt werden, kann die Wissensbasis auch die nötigen Informationen zum Erstellen von Formularen bereitstellen. Um ein Formular zu einem Dokumenttypen zu erstellen, mit dessen Hilfe Dokumente dieses Dokumenttyps erfasst werden können, wird von der Wissensbasis die Struktur des jeweiligen Dokumenttyps inklusive der jeweils zugelassenen Ausprägungen, Kardinalitäten etc. zurückgeliefert. Dieselbe Information ist auch für das Erzeugen von Validierungsskripten wie etwa XML-Schemata notwendig. Die Wissensbasis kann zusätzlich zur Prüfung der Konsistenz des darin enthaltenen Wissens herangezogen werden.

### **3.1.2 Die inhaltsbasierte Suchanfrage**

Eine inhaltsbasierte Suchanfrage kann als Erweiterung zur metadatenbasierten Suche gesehen werden. Neben den Informationen zum Patienten, dem gewünschten Zeitbereich oder den Dokumenttypen können bei der inhaltsbasierten Suche im Gegensatz zur metadatenbasierten Suche noch zusätzlich Suchbegriffe, die sich auf den medizinischen Inhalt beziehen, angegeben werden.

Die erlaubten Suchbegriffe werden aus der Wissensbasis zur Verfügung gestellt um eine thesaurusbasierte Suche umzusetzen. Im Gegensatz zu einer Freitextsuche werden bei einer thesaurusbasierten Suche nur Suchbegriffe aus einem vordefinierten Pool (i.e. Thesaurus) angeboten. Alle medizinischen Konzepte, die in der Wissensbasis abgebildet sind, werden als mögliche Suchbegriffe angeboten. Die so erzeugte thesaurusbasierte Suche ist genau an die dokumentierten Inhalte angepasst. Die Suchbegriffe in den Suchanfragen müssen nicht auf orthographisch ähnliche Wörter, Synonyme oder Rechtschreibfehler untersucht werden, was eine Verarbeitung der Suchanfrage erleichtert.

Typische Zeiteinschränkungen werden beim Erstellen der Suchanfrage zur Verfügung gestellt. Dies erlaubt es, Suchanfragen schneller zu formulieren, da zum Beispiel direkt die Zeiteinschränkung "Dokumente der letzten sechs Monate" gewählt werden kann. Zu jedem Suchbegriff kann ein eigener Zeitbereich angegeben werden.

Die von der Wissensbasis zur Verfügung gestellten Begriffe können in separaten Suchanfragen

abgesetzt oder mittels booleschen Operatoren zu komplexeren Suchanfragen zusammengesetzt werden. Die Suchanfragen, die aus mehreren Begriffen formuliert werden, erlauben es dem GDA einen Überblick über mehrere Gesundheitsparameter eines Patienten zu bekommen. Diese Umsetzung ermöglicht es unter anderem auch, Suchanfragen zu erzeugen, die eine *Patient Summary*, wie sie etwa im *EPSOS*-Projekt verwendet wird, aus existierenden Dokumenten als Suchergebnis zurückliefern. Die Verbindung der jeweiligen Begriffe wird beim Erzeugen der metadatenbasierten Suche und der XQuery-Suchanfrage mithilfe der Wissensbasis automatisch hergestellt.

Die erreichbare Granularität einer Suchanfrage hängt vom Strukturierungsgrad des in der Wissensbasis abgebildeten Wissens ab. Dies hat den Vorteil, dass dem Arzt kein Detaillierungsgrad der Suche vorgetäuscht wird, der durch die Dokumente nicht direkt erfüllbar ist. Ist der Wunsch nach detaillierteren Suchen in spezifischen medizinischen Gebieten vorhanden, muss der Strukturierungsgrad der zugrundeliegenden Dokumenttypen angepasst werden.

Neben den jeweiligen Suchbegriffen werden in der inhaltsbasierten Suchanfrage die ID des Patienten, dessen Dokumente gesucht werden, sowie die ID des GDA, der die Suche ausführt, als Metadaten übergeben.

### **3.1.3 Die metadatenbasierte Suchanfrage**

Die bei den GDAs abgelegten EHR-Dokumente werden in der zentralen Metadatenkomponente registriert, sodass andere GDAs diese mithilfe einer metadatenbasierten Suchanfrage an die Metadatenkomponente finden können. Bei der metadatenbasierten Suchanfrage können unter anderem die jeweiligen Dokumenttypen, die Patienten-IDs sowie die Zeitbereiche für eine Suche von relevanten Dokumenten eingeschränkt werden. Die Metadaten in der Metadatenkomponente sind unabhängig vom zugrundeliegenden Dokumenttyp und dem Standard, in dem das Dokument erstellt wurde, registriert. EHR-Dokumente können so unabhängig von der Struktur übermittelt werden (i.e. PDF-Dokumente, ISO/EN-13606-EHR-Extrakte, CDA-Dokumente etc.). Als Antwort auf eine metadatenbasierte Suche liefert die Metadatenkomponente die Metadaten der potenziell relevanten Dokumente sowie deren Speicherort beim jeweiligen GDA zurück. Die anhand der Metadaten für den GDA relevant erscheinenden EHR-Dokumente können dann direkt von dem jeweiligen GDA

heruntergeladen werden.

Diese metadatenbasierte Suchanfrage wird mithilfe der Wissensbasis und der Werte in der inhaltsbasierten Suchanfrage automatisch erstellt. Die ID des Patienten kann direkt aus der inhaltsbasierten Suchanfrage übernommen werden. Aus den Zeitbereichen, die den jeweiligen inhaltlichen Suchbegriffen zugeordnet sind, werden die Zeitbereiche für die metadatenbasierte Suche generiert. Aus den inhaltlichen Suchbegriffen werden die Dokumenttypen berechnet. Wie in Abbildung 3.2 ersichtlich, ist jeder inhaltliche Suchbegriff über die Wissensbasis und die Archetypen mit einem EHR-Inhalt in einem Dokumenttyp verbunden. Dadurch ist es möglich jeden inhaltlichen Suchbegriff einem oder mehreren Dokumenttypen zuzuordnen.

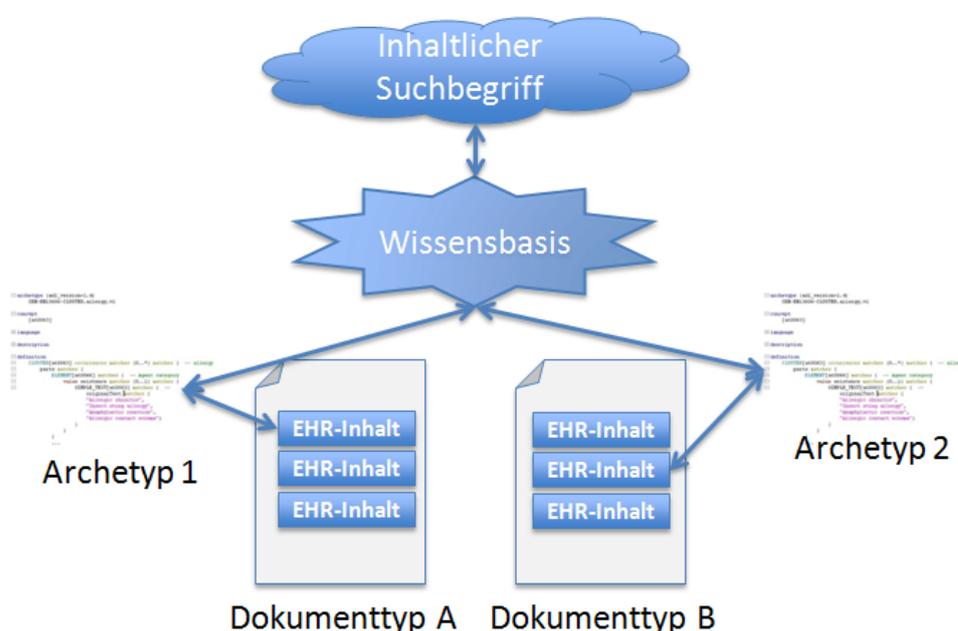


Abbildung 3.2: Zusammenhang inhaltlicher Suchbegriff, Wissensbasis, Archetyp und Dokumenttyp.

### 3.1.4 Die XQuery-Suchanfrage

Um in den EHR-Dokumenten nach den gewünschten Begriffen zu suchen, muss die inhaltsbasierte Suchanfrage in einen für die "Suchmaschine" interpretierbaren Ausdruck umgewandelt werden. Dieser Schritt wird in [95] auch als *Query Binding* bezeichnet. Im Rahmen des Projekts EHR-Arche werden EHR-Dokumente, die als XML-Dokumente abgelegt

sind, durchsucht. Die inhaltsbasierte Suchanfrage wird daher in eine XQuery-Suchanfrage umgewandelt, die direkt auf die EHR-Dokumente abgesetzt werden kann.

Das Erzeugen der XQuery-Suchanfrage kann in folgende Teile unterteilt werden:

- Auflösen der Suchbegriffe
- Bestimmen der *Minimal Information Unit*
- Boolesche Operatoren auflösen
- XQuery-Suchanfrage erzeugen

#### **3.1.4.1 Auflösen der Suchbegriffe**

Zu allen in der inhaltsbasierten Suchanfrage angegebenen inhaltlichen Suchbegriffen werden die jeweiligen Knoten in den Dokumenttypen gesucht. In der Wissensbasis sind jedem Suchbegriff medizinische Inhalte in gewissen Dokumenttypen zugewiesen. Wie zum Beispiel bei einem Laborparameter, der sowohl in EHR-Dokumenten vom Typ "Arztbrief" als auch in EHR-Dokumenten vom Typ "Laborbefund" vorkommt, kann ein Suchbegriff mehreren Knoten zugewiesen werden. Jeder Suchbegriff ist aber mindestens einem konkreten Knoten in den Dokumenttypen zugewiesen.

#### **3.1.4.2 Bestimmen der Minimal Information Unit**

Bevor die inhaltsbasierte Suchanfrage in eine XQuery-Suchanfrage umgewandelt werden kann, muss die *Minimal Information Unit* [133] der Knoten bestimmt werden. Die *Minimal Information Unit* beschreibt die Granularität einer Information, sodass diese in sich abgeschlossen, aussagekräftig und gleichzeitig spezifisch ist. Die *Minimal Information Unit* wird benötigt, um *focused retrieval* (siehe Kapitel 2.4.2) zu ermöglichen.

Aus den im vorherigen Schritt über die Suchbegriffe gefundenen Knoten können Rückschlüsse auf die *Minimal Information Unit* getroffen werden. Bezieht sich ein Suchbegriff auf Knoten, die einem Dokumenttyp oder einer Beschreibung einer ganzen klinischen Behandlung zugeordnet werden können, wird der Knoten direkt übernommen. Für Knoten, die nur einen einzelnen Wert beschreiben, muss geprüft werden, ob der Wert alleine genug Aussagekraft besitzt. Handelt es sich zum Beispiel um einen Wert aus einer Tabelle, muss der Spaltenname zusätzlich angegeben

werden. Bezieht sich der Wert nicht auf den Patienten selber, sondern ein Familienmitglied, wie dies z.B. im Arztbrief bei der Familienanamnese der Fall sein kann, darf dieser Kontext nicht verloren gehen. Bildet ein Knoten keine *Minimal Information Unit*, wird diese berechnet und der Knoten durch diesen neu berechneten Knoten ersetzt.

### 3.1.4.3 Boolesche Operatoren auflösen

Bei einer Verknüpfung von ODER und UND muss klar festgelegt werden, wie die Operatoren auf eine Suche angewendet werden und was zurückgegeben werden soll. Anhand des Beispiels in Abbildung 3.3 wird erklärt, wie UND und ODER in der vorliegenden Arbeit aufgelöst werden.

Bei ODER-Verknüpfungen wird davon ausgegangen, dass keine direkte Beziehung zwischen den verknüpften Suchbegriffen vorhanden ist, d.h. die Suchbegriffe werden unabhängig voneinander behandelt. Wird in Abbildung 3.3 “Knoten 4 ODER 5” gesucht, werden zwei Container zurückgeliefert, die jeweils die Knoten 4 und 5 mit den jeweiligen Unterbäumen enthalten. Eine Suche nach “Blutdruckmessung” ODER “Körpergewicht” liefert einen Container mit allen “Blutdruckmessungen” und einen mit allen “Körpergewichten” zurück.

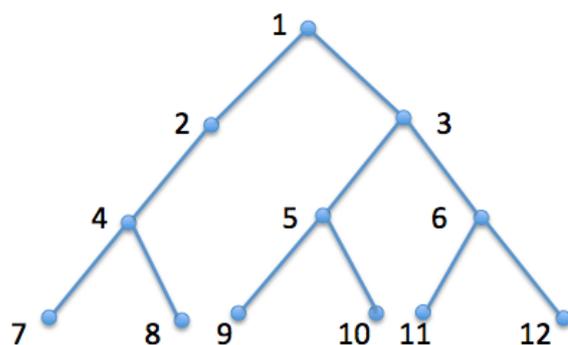


Abbildung 3.3: Auflösen der UND- und ODER-Verknüpfung anhand einer exemplarischen Baumstruktur.

Etwas anders wird die Suche zwischen UND-verknüpften Suchbegriffen aufgelöst. Bei einer UND-Verknüpfung wird davon ausgegangen, dass beide verknüpften Suchbegriffe im selben Dokument vorhanden sind. Das Dokument wird als größter gemeinsamer Container angesehen, UND-Verknüpfungen über mehrere Dokumente gleichzeitig werden nicht berücksichtigt. Durch das Verwenden einer UND-Verknüpfung wird beim Erzeugen der inhaltsbasierten

Suchanfrage automatisch eine Beziehung zwischen den Suchbegriffen angenommen. Um dies in der XQuery-Suchanfrage abzubilden, wird bei einer UND-Verknüpfung der gemeinsame Elternknoten der UND-verknüpften Knoten gesucht. Wird in Abbildung 3.3 nach 7 UND 8 gesucht, ist der gemeinsame Elternknoten 4. Dieser wird in einem Container als Antwort zurückgeliefert. Wird nach Knoten 12 UND 4 UND 5 gesucht, ist der gemeinsame Elternknoten 1, von Knoten 9 UND 11 ist es der Knoten 3.

Das Zurückliefern des gemeinsamen Elternknotens bei UND-Verknüpfungen ermöglicht es, den Zusammenhang von zwei Suchbegriffen abzubilden. Der Nachteil dieser Methode ist, dass bei UND-Verknüpfungen keine feingranularen Suchergebnisse zurückgeliefert werden, aus denen beispielsweise direkt ein Graph zur übersichtlichen Darstellung generiert werden kann. Wie auch beim Finden der *Minimal Information Unit* wird die Suchanfrage so angepasst, dass die gemeinsamen Elternknoten berücksichtigt werden.

#### **3.1.4.4 XQuery-Suchanfrage erzeugen**

Um aus der aufgelösten Suchanfrage eine XQuery zu erzeugen, wird die Suchanfrage in eine disjunktive Normalform transformiert. Diese Transformation erlaubt es, aus jeder entstandenen Konjunktion direkt einen XQuery-FLOWR-Ausdruck (siehe Kapitel 2.2) zu erzeugen.

#### **3.1.5 Aufbereiten der Suchergebnisse**

Das Ergebnis der XQuery muss im letzten Schritt aufbereitet werden, bevor es dem GDA zurückgeliefert wird. Das Ergebnis der XQuery ist ein XML-Dokument, das abhängig von der XQuery strukturiert ist und die relevanten Teile der durchsuchten EHR-Dokumente enthält.

Für eine tabellarische Visualisierung wird für jede Konjunktion der in disjunkten Normalform aufgelösten Suchanfrage eine Zeile in der Rückgabetable reserviert (siehe Abbildung 3.4). In der ersten Spalte werden die Beschreibungen der jeweiligen Zeilen angegeben. Die Ergebnisse der XQuery-Suchanfrage werden in chronologischer Reihenfolge abgearbeitet und die Werte in die entsprechenden Zeilen eingefügt. In der so erzeugten Tabelle entspricht jede Zelle einem Suchergebnis einer Konjunktion aus einem Dokument. Die untereinander liegenden Zellen einer Spalte können dabei von unterschiedlichen Dokumenten stammen. Die aktuellsten Werte stehen immer auf der linken Seite der Tabelle, Lücken gibt es keine.

Die einzelnen Konjunktionen können zusätzlich aufgeteilt werden, um einzelne Werte besser

Beschreibung	1. Ergebnis	2. Ergebnis	...
	K1, E1	K1 E2	...
	K2, E1	K2 E2	
	...	...	

Abbildung 3.4: Tabellarische Visualisierung der Konjunktionen der disjunkten Normalform.

vergleichen zu können. Dazu werden für alle elementaren Werte die Zeilen in weitere Zeilen geteilt und die Werte in die richtigen Zeilen geschrieben. Ein Beispiel einer solchen Tabelle ist in Abbildung 5.9 auf Seite 130 zu sehen.

### 3.2 Iterativer Archetypen-Entwicklungszyklus

Um eine inhaltsbasierte Suche durchzuführen, müssen die EHR-Dokumente mit Archetypen beschrieben sein. Im Zwei-Modell-Ansatz werden Information und Wissen strikt voneinander getrennt. Wissen wird mittels Archetypen modelliert und kann so als Basis für eine Dateneingabe oder einen Datenaustausch in EHR-Systemen verwendet werden.

Für die Erzeugung der Archetypen wurden unterschiedliche Tools und Editoren entwickelt. Einer der ersten Archetyp-Editoren war der *openEHR Archetype Editor* [134]. Der Editor ist speziell auf openEHR-Archetypen zugeschnitten, in früheren Versionen unterstützte der Editor auch ISO/EN-13606-Archetypen. Die Oberfläche wurde in mehrere Sprachen übersetzt und der Source Code freigegeben, der Editor ist jedoch nur für Windows erhältlich. In [42] wird der betriebssystemunabhängige LIU Archetypen Editor, der in Java programmiert ist, vorgestellt. Der Funktionsumfang ist fast ident mit dem openEHR Archetype Editor, wobei versucht wird die Eingabe von Terminologien über den *term\_binding*-Mechanismus in Archetypen sowie die Usability zu verbessern. Der Archetyp-Editor von *medical-objects* [135] unterstützt ISO/EN-13606-Archetypen und bietet wie auch der LIU Archetyp Editor eine Voransicht eines möglichen Formulars basierend auf dem jeweiligen Archetyp. In dieser können jedoch keine Daten eingegeben werden. Der *LinkEHR* Editor [136] unterstützt sowohl openEHR-, ISO/EN-13606- und CDA-Archetypen. Da die CDA-Spezifikation nicht öffentlich zugänglich ist, muss das CDA-Schema nach dem Download separat von HL7 bezogen werden. Der

*LinkEHR* Archetyp-Editor wird in der vorliegenden Arbeit zum Erzeugen der Archetypen verwendet.

Erfahrungen im Projekt EHR-Arche haben gezeigt, dass zusätzlich zu den Archetyp-Editoren Methoden entwickelt werden müssen, um die GDAs beim Erzeugen der Archetypen stärker einzubinden. Für die Erzeugung von Archetypen wird in der vorliegenden Arbeit das iterative Software-Entwicklungs-Modell von [137] angewandt. Der Ansatz von Garde et al. [132] wird erweitert, in dem eine iterative Verfeinerung der Archetypen umgesetzt wird und Mediziner stärker beim Erstellen der Archetypen eingebunden werden.

In Abbildung 3.5 wird der im Rahmen der vorliegenden Arbeit erstellte Iterative Entwicklungszyklus für Archetypen dargestellt [138]. Basierend auf den Anforderungen der GDAs und existierenden Archetypen wird die initiale Planung gemacht. Diese Information wird verwendet, um Archetypen zu erzeugen und anzupassen. Aus diesen Archetypen wird im Gegensatz zu der Voransicht eines möglichen Formulars ein funktionsfähiges Formular generiert, das für das Evaluieren der Archetypen verwendet werden kann. Basierend auf den Resultaten der Evaluierung werden die gewünschten Änderungen geplant und der Zyklus wird iteriert, bis schlussendlich die fertigen Archetypen freigegeben werden.

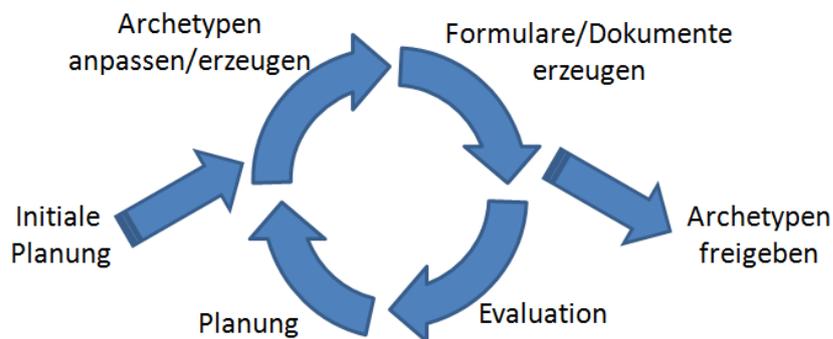


Abbildung 3.5: Überblick des iterativen Entwicklungszyklus für Archetypen.

### 3.2.1 Initiale Planung

Die initiale Planung umfasst die ersten zwei Schritte des Ansatzes in [132]. Im ersten Schritt “Define high level archetype concepts” werden die in dem Archetyp abzubildenden medizinischen Inhalte und Strukturen beschrieben und hierarchisch strukturiert. Die Beschreibung findet auf einer höheren Ebene statt, daher sollte nicht die ADL zur

Dokumentation dieser Ergebnisse herangezogen werden, sondern eher Mind-Maps oder Tabellen. Um die enge Zusammenarbeit der GDAs und Techniker in diesem Schritt zu unterstützen, sollte ein Dokumentationsformat gewählt werden, mit dem alle Beteiligten arbeiten können.

Im zweiten Schritt “Check for existing reference archetypes” werden existierende Archetypen gesucht, welche die im ersten Schritt beschriebenen medizinischen Inhalte abbilden. Dieser Schritt dient dazu, den Wildwuchs an Archetypen zu verhindern und die semantische Interoperabilität zwischen EHR-Systemen, die auf den existierenden Archetypen aufbauen, zu ermöglichen. Als Ausgangspunkt können zusätzlich zu Archetypen auch *Detailed Clinical Models*, Templates, existierende Implementierungsleitfäden, Nachrichtenspezifikationen, medizinische Terminologien oder existierende EHR-Systeme dienen, die ähnliche Daten beschreiben. Die aufgelisteten Quellen haben meist schon mehrere Überarbeitungszyklen durchlaufen und können unter anderem Hinweise über existierende Anwendungsmuster und Anwendungsszenarien geben oder Details zeigen, die beim Definieren der Konzepte, die in den Archetypen abzubilden sind, noch nicht berücksichtigt wurden.

### **3.2.2 Erzeugung und Anpassen der Archetypen**

Basierend auf der Planung werden neue Archetypen erzeugt oder bestehende Archetypen spezialisiert. Wie in [132] empfohlen wird ein visueller Archetyp-Editor verwendet. Das Erzeugen und Anpassen der Archetypen wird vom Informatiker gemacht, da die in Archetypen verwendeten Konzepte wie Vererbung oder Kardinalitäten für GDAs eher nicht vertraut sind. Die medizinische Qualität der Archetypen wird durch die im vorherigen Schritt erfolgte gemeinsame Planung der GDAs und Techniker gesichert.

### **3.2.3 Generierung der Formulare und Dokumente**

Die in [132] beschriebene Methode wird in diesem Schritt erweitert. Mithilfe eines Tools zur automatischen Formulargenerierung aus Archetypen (siehe Kapitel 3.3) können archetypkonforme EHR-Dokumente erzeugt werden. Den GDAs werden statt des abstrakten Konzepts der Archetypen vertraute Konzepte wie Formulare und Dokumente zur Verfügung gestellt. Fehlende Teile, falsche hierarchische Strukturen oder falsche Kardinalitäten können damit durch eine geringere Einarbeitungszeit erkannt werden.

In diesem Schritt wird der Archetyp zusätzlich auf seine formale Korrektheit überprüft. Durch die Verwendung von Archetyp-Editoren wird sichergestellt, dass der Archetyp syntaktisch gemäß der ADL-Spezifikation erstellt ist und der Archetyp den Klassen des Referenzmodells entspricht. Bei der Formularerzeugung kann zusätzlich überprüft werden, ob alle benötigten Archetypen, die über *Slots* eingebunden sind, auch vorhanden sind und ob die Namen, Übersetzungen und Terminologie-Codes der unterschiedlichen Archetyp-Knoten richtig vergeben sind.

### **3.2.4 Evaluierung und Planung**

Die Evaluierung und die weitere Planung führen zu weiteren Anpassungen der Archetypen, wodurch ein Iterationszyklus abgeschlossen ist. Die Evaluierung basiert auf den Erkenntnissen der GDAs, die durch die Begutachtung der automatisch generierten Formulare und die damit erzeugten Dokumente ermittelt werden. Zusätzlich können Fehler, die durch die computergestützte Verarbeitung der Archetypen erkannt werden, behoben werden. Diese Erkenntnisse dienen als Basis für die weitere Planung, womit ein neuer Iterationszyklus beginnt. Wie bei der initialen Planung werden auch hier die zu ändernden Konzepte zwischen Technikern und GDAs abgesprochen, bevor die Archetypen angepasst und eventuell neue Archetypen erzeugt werden.

### **3.2.5 Freigabe der Archetypen**

Sind keine Änderungen mehr erforderlich, können die Archetypen freigegeben werden. Für einen semantisch interoperablen Datenaustausch ist es wichtig, dass alle beteiligten GDAs und EHR-Systeme Zugriff auf die Archetypen haben. Für openEHR existiert ein *Clinical Review Board* und ein öffentlich zugängliches Archetyp-Repository, in dem Archetypen veröffentlicht werden. Für ISO/EN 13606 gibt es keine solche zentrale Infrastruktur, sie könnte in Zukunft aber von der *EN 13606 Association* [32] angeboten werden. Für den CDA-Standard wurden bis jetzt keine Archetypen erzeugt. CDA-Implementierungsleitfäden werden über einen Ballot von den nationalen HL7 Affiliates als Teil von HL7 aufgenommen, dieser Ablauf könnte auch für Archetypen übernommen werden. CDA-Archetypen wären in diesem Fall für Mitglieder über die HL7-Homepage zugänglich.

### 3.3 Plug-and-Play-Formulargenerierung

Um eine inhaltsbasierte Suche basierend auf Archetypen zu testen, müssen archetypkonforme Dokumente zu Testzwecken vorhanden sein. Im Rahmen des EHR-Arche-Projektes wurden ISO/EN-13606-EHR-Extrakte verwendet, die bis auf ELEMENT-Ebene durch Archetypen beschrieben sind. In Kapitel 2.1.3.1 werden Ansätze zum Erzeugen von archetypkonformen EHR-Dokumenten beschrieben. In [54] präsentieren wir einen Ansatz zur Integration von Archetypen in ein bestehendes EHR-System, um die Daten des Systems in weiterer Folge als ISO/EN-13606-EHR-Extrakte zu exportieren.

Im Folgenden wird die von Kohler und Rinner entwickelte Methode zur Generierung von Formularen aus Archetypen sowie die anschließende Erzeugung von archetypkonformen ISO/EN-13606-EHR-Extrakten aus [91, 139] beschrieben und durch relevante Teile aus [54] ergänzt. Von Kohler stammen hauptsächlich die allgemeinen softwaretechnischen Implementierungen wie etwa die Umsetzung des MVC-Pattern und die Schnittstellenspezifikation. Die Archetypen und EHR-spezifischen Teile stammen hauptsächlich vom Autor der vorliegenden Arbeit. Die Methode wurde von Kohler et al. [139, 91] bei zwei Konferenzen präsentiert und bei der Konferenz *E-Health 2011* mit dem Dissertanten-Award ausgezeichnet.

Das *Model-View-Controller*(MVC)-Pattern von Reenskaug [140] wird verwendet, um eine Wiederverwendung und ein Herauslösen der einzelnen Komponenten zu ermöglichen.

Ein Überblick über die Zusammenhänge zwischen *Model*, *View* und *Controller* ist in Abbildung 3.6 gegeben. Die Methode ist an die Anforderungen des Projekts EHR-Arche angepasst, in dem eine dezentrale Speicherung der EHR-Dokumente vorgesehen ist und eine Wissensbasis das nötige Wissen zur Generierung von Formularen bereitstellt. Aufbauend auf dem Wissen der Wissensbasis (i.e. Struktur der Archetypen und des Referenzmodells) wird das *Model* aufgebaut. Mithilfe der *View* werden Formulare für das jeweilige *Model* dargestellt. Der *Controller* ist für das Erzeugen des *Model* und der *View* sowie die Verarbeitung der Eingaben der Benutzer sowie das Erzeugen der EHR-Dokumente zuständig.

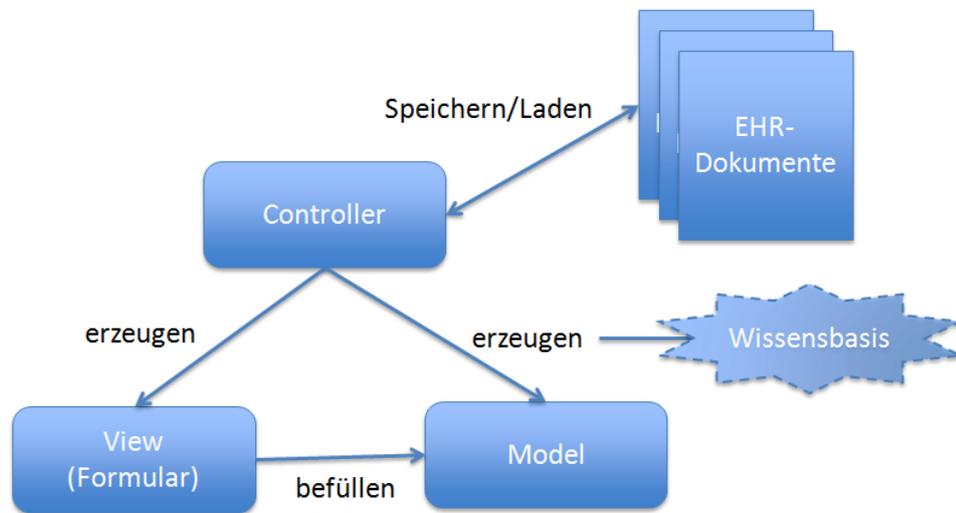


Abbildung 3.6: Zusammenhang zwischen *Model*, *View* und *Controller*.

### 3.3.1 Die Model-Komponente

Zwei unterschiedliche Ansätze für das *Model* können unterschieden werden. Es kann von einem existierenden Modell eines existierenden EHR-Systems ausgegangen werden oder ein speziell für die Vorgaben des jeweiligen EHR-Standards optimiertes *Model* herangezogen werden. In [54] wird von einem proprietären Modell eines existierenden EHR-Systems ausgegangen. Es muss dabei beachtet werden, dass eine sogenannte “modelling safe zone” [53] gefunden wird, in der das Modell des EHR-Standards und das existierende EHR-System-Modell überlappen. Je nach Größe der Überlappung können mehr oder weniger Konzepte aus dem EHR-Modell ins bestehende EHR-System übernommen werden. Ein für den EHR-Standard optimiertes Modell sollte vollständig mit dem EHR-Modell überlappen.

Im Rahmen des EHR-Arche-Projektes wird das *Archetype Object Model* (AOM) als Grundlage für das *Model* verwendet. Das *Model* ist in diesem Fall an die Vorgaben des EHR-Standards angepasst, da im AOM standardmäßig Archetypen abgebildet werden. Da Archetypen, die Instanzen des AOM, nur jene Attribute beinhalten, die vom Referenzmodell abweichen, sollte das Konzept des *comprehensive Archetype* von [136] verwendet werden. Der *comprehensive Archetype* vereint die Information des Referenzmodells und das Wissen des Archetyps in einem sogenannten *comprehensive Archetype*, der ebenfalls als Instanz des AOM abgebildet wird. Dieser beinhaltet die gesamte Klassenhierarchie des Referenzmodells mit allen Attributen und

Relationen, auch wenn nur ein Teil davon vom Archetyp eingeschränkt wird. Im *comprehensive Archetype* werden auch die Archetypen, die über *Slots* eingebunden werden, dargestellt. Diese dynamische Ableitung kann als temporärer Wechsel vom Zwei-Modell-Ansatz in einen Gesamtmodellansatz gesehen werden, um Aufgaben wie Erzeugen von EHR-Daten, Validieren oder Mappen zwischen EHR-Standards zu erleichtern. Das AOM ist ein speziell für die Bedürfnisse der Archetypen entwickeltes Modell und unterstützt alle in Archetypen umgesetzten Funktionen. Die “modelling safe zone” ist durch die vollkommene Überlappung der beiden Modelle gegeben. Das Verwenden des *comprehensive Archetype* stellt sicher, dass auch sämtliche benötigte Information aus dem Referenzmodell vorhanden ist.

Um Formulare mithilfe des AOM darzustellen, muss das AOM um folgende Funktionen erweitert werden:

- Im AOM müssen die im Formular dokumentierten Daten zu Archetypenknotten abgelegt werden können.
- Archetypenknotten, für welche der Archetyp eine Kardinalität  $>1$  vorsieht, müssen im AOM verdoppelt und gelöscht werden können.
- Jeder Knoten muss eindeutig referenzierbar sein, die Verdoppelung von Archetypknotten darf nicht zu mehrdeutigen Knoten-IDs führen.

### **3.3.2 Die View-Komponente**

Die *View* umfasst jene Komponenten, die vom Benutzer gesehen werden, sie kann mit einem Formular verglichen werden. Um das Formular darzustellen, werden aus dem *Model*, das aus dem AOM und dem *comprehensive Archetype* erzeugt wird, generisch die unterschiedlichen Eingabefelder erzeugt. Die jeweiligen Eingabemöglichkeiten hängen von den im Referenzmodell verwendeten Datentypen und den jeweiligen Klassen ab.

Als Ausgangspunkt für die Erzeugung des Formulars muss ein konkreter Knoten im *comprehensive Archetype* angenommen werden. Im Falle des Standards ISO/EN 13606 kann die COMPOSITION-Klasse als Ausgangspunkt herangezogen werden, bei der HL7 CDA das *clinical statement*. Darüber liegende Informationen zum Autor, Patienten etc. werden nicht generisch vom Formular abgebildet, sondern über statische Eingabefelder realisiert. Neben dem

Formular selber müssen Komponenten für die Formularblätter, Gruppierungen, Tabellen und einzelnen Werte festgelegt werden.

Formulareingaben in der *View* werden als Instanzen des *Model* gespeichert.

### **3.3.3 Die Controller-Komponente**

Im *Controller* wird der *comprehensive Archetype* erzeugt, um daraus das *Model* abzuleiten und das Formular zu erzeugen.

Durch das Verwenden des erweiterten AOM als *Model* kann das EHR-Dokument direkt aus den Formulareingaben generiert werden.

Im *Controller* werden die existierenden EHR-Dokumente zu einem Patienten verwaltet. Sie können geladen, gespeichert, bearbeitet und in einer dezentralen Datenhaltung zur Verfügung gestellt werden.

## **3.4 XML-Schema zur semantischen Validierung**

In diesem Kapitel werden die Kernpunkte des vom Autor dieser Arbeit publizierten Journalartikels zur semantischen Validierung von ISO/EN-13606-EHR-Extrakten [141] zusammengefasst. Bevor EHR-Dokumente ausgetauscht oder in einem Shared-EHR-System wie IHE-XDS zur Verfügung gestellt werden, sollten die Dokumente validiert werden, um eine einwandfreie Verarbeitung im Zielsystem sicherzustellen [61]. Dies ist speziell bei XML-Dokumenten wichtig, da diese von unterschiedlichsten Anwendungen generiert werden können. Proprietäre Dateiformate können im Gegensatz dazu meist nur von einer einzigen Software erzeugt werden, wodurch ein einheitliches Format der Dokumente von vornherein gegeben ist. Im vierten Teil des Standards ISO/EN 13606 wird die Integrität der EHR-Dokumente, die ausgetauscht, gespeichert oder verarbeitet werden, als wesentliche Sicherheitsanforderung angegeben. Diese kann nur durch ein Validieren der Daten erreicht werden.

Für die syntaktische Validierung (siehe Kapitel 2.1.3.2) von EHR-Dokumenten wird in dem in [141] vorgeschlagenen Ansatz XML-Schema [75] verwendet. Für die Referenzmodelle der CDA und OpenEHR werden offizielle XML-Schemata zur Verfügung gestellt. Für die ISO/EN 13606 existiert nur ein inoffizielles XML-Schema der *EN 13606 Association* [32]. Neben XML-

Schema existiert ein Tool basierend auf Eclipse, um CDA-Dokumente syntaktisch zu validieren [142].

Im Folgenden wird der in [141] vorgestellte Ansatz zusammengefasst, um XML-Schema auch für die semantische Validierung (siehe Kapitel 2.1.3.2) zu verwenden. Die Regeln, die im Archetyp vorgegeben sind, werden mit XML-Schema überprüft. Die *Unique Particle Attribution Constraint Rule* in XML-Schema besagt, dass auf einer hierarchischen Ebene in XML-Schema nicht mehrere gleichnamige XML-Elemente mit unterschiedlichen Unterstrukturen beschrieben werden können. Da sich im Zwei-Modell-Ansatz die generischen Klassen jedoch wiederholen (siehe z.B. die zwei ELEMENTE, für “lab\_test\_result” und “lab\_test\_reference\_range” auf derselben Ebene in Listing 2.1 auf Seite 19), ist ein Wiederholen von gleichnamigen XML-Elementen auf einer hierarchischen Ebene notwendig und wird mit dem in [143, 141] vorgestellten Ansatz umgesetzt.

Neben XML-Schema existieren noch andere Sprachen wie Schematron [57] oder RELAX NG [144] zur Validierung von XML-Dokumenten. Rinner et al. konzentrieren sich jedoch in [141] auf XML-Schema, da folgende Vorteile dadurch erwartet werden:

XML-Schema wird schon für die syntaktische Validierung im EHR-Umfeld verwendet. Der gesamte Validierungsablauf könnte vereinfacht werden, wenn die gleiche Technologie für die semantische Validierung herangezogen wird, im Gegensatz zur Verwendung einer zusätzlichen Technologie wie Schematron oder RELAX NG. Der hierarchische Aufbau von XML-Schema erlaubt es, die vorgegebenen Strukturen in Archetypen gut darzustellen. Aus Schematron-Regeln ist die hierarchische Struktur des EHR-Dokuments im Vergleich dazu nicht direkt ersichtlich, da immer nur Teilstrukturen mit einer Regel dargestellt werden. In [18] wird ein Ansatz präsentiert, um proprietäre EHR-Daten in standardisierte ISO/EN-13606-EHR-Extrakte umzuwandeln. Die Transformationsregeln werden als Mapping zwischen XML-Schematas dargestellt. Dasselbe XML-Schema kann zusätzlich für die semantische Validierung herangezogen werden. XML-Schema ist außerdem ein offizieller W3C-Standard. Dadurch existieren unzählige Tools für verschiedenste Betriebssysteme und Programmiersprachen, darunter auch Open-Source-Implementierungen. Dasselbe gilt auch für XSLT-Skripte, die aus Schematron-Regeln generiert werden können. XSLT ist jedoch hauptsächlich für die Transformation von XML-Dokumenten konzipiert und nicht deren Validierung. Unabhängig davon würde man für eine Schematron-Validierung statt XSLT eher

native Tools heranziehen, wobei diese Tools nicht im selben Ausmaß wie für XML-Schema angeboten werden. Weiters kann durch die Verwendung von XML-Schema der Einstieg in die EHR-Welt für Nutzer erleichtert werden, die kein Vorwissen über Archetypen und den Zwei-Modell-Ansatz besitzen, aber mit XML-Technologien vertraut sind.

Die semantische Validierung nach Rinner et al. gliedert sich in folgende zentrale Schritte:

1. Um die *Unique Particle Attribution Constraint Rule* von XML-Schema zu erfüllen, wird aus dem Referenzmodell und dem Archetyp ein gemeinsames XML-Schema mit einer speziellen Namensgebung erzeugt.
2. Das EHR-Dokument wird mittels eines generischen XSLT-Skripts auf die gleiche Namensgebung transformiert.
3. Das transformierte EHR-Extrakt wird mit dem XML-Schema semantisch validiert (siehe Abbildung 3.7).

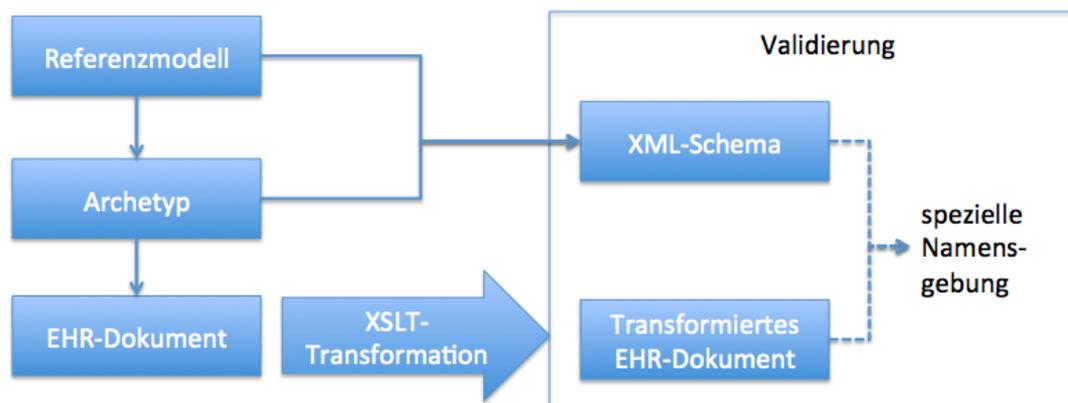


Abbildung 3.7: Semantische Validierung mittels XML-Schema, adaptiert von [141].

### 3.4.1 Erfüllung der *unique particle attribution constraint rule* von XML-Schema

Beim Zwei-Modell-Ansatz werden Klassen des Referenzmodells von einem Archetypen eingeschränkt. Dies resultiert oft in mehreren Instanzen derselben Referenzmodellklasse auf derselben hierarchischen Ebene. In Abbildung 3.8 wird ein Beispiel dieser Problematik anhand einer UML-Darstellung des ISO/EN-13606-Referenzmodells auf der linken Seite dargestellt

sowie einem Archetyp in ADL in der Mitte, die als Vorlage für das EHR-Extrakt auf der rechten Seite dienen. Die beiden gezeigten SECTION-Klassen auf derselben hierarchischen Ebene mit unterschiedlichem Inhalt können durch die *unique particle attribution constraint rule* nicht direkt mit XML-Schema dargestellt werden.

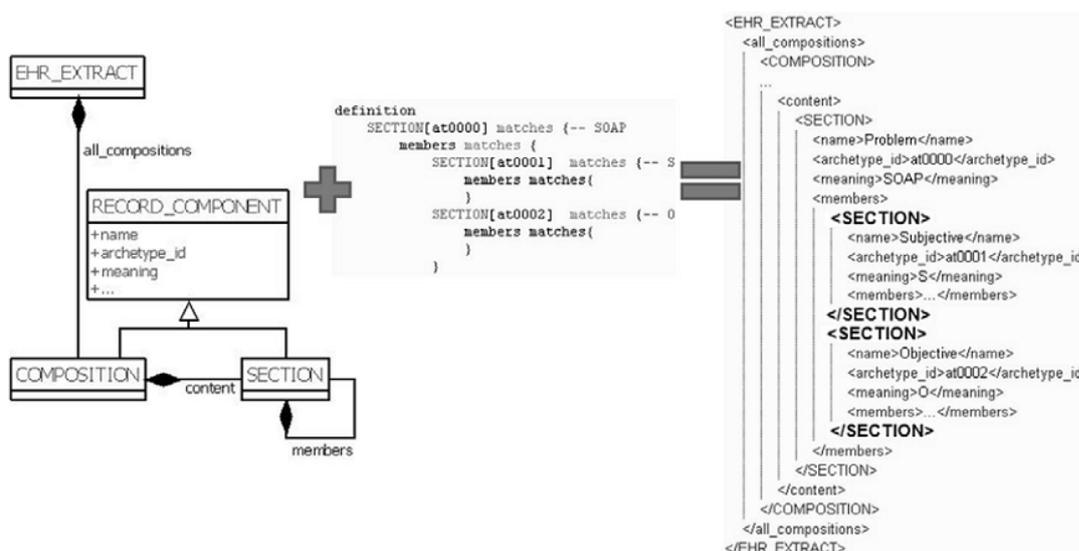


Abbildung 3.8: Beispiel zur *Unique Particle Attribution Constraint Rule* von XML-Schema, adaptiert von [141].

Durch eine spezielle Namensgebung der XML-Schema-Elemente, die durch diese Probleme betroffen sind, kann dieses Problem umgangen werden. Dabei wird der ursprüngliche Name der betroffenen Elemente um die zugrundeliegende Archetyp-Knoten-ID des sich wiederholenden Elements erweitert. Die beiden SECTION-Instanzen aus Abbildung 3.8 werden im XML-Schema mit SECTION\_at0001 und SECTION\_at0002 bezeichnet.

### 3.4.2 Erzeugen des XML-Schemas

Um ein EHR-Dokument semantisch zu validieren, müssen die Informationen des Referenzmodells und die Vorgaben, die in den Archetypen gemacht werden, gleichzeitig berücksichtigt werden. Für jeden Archetyp wird auf Basis der zuvor beschriebenen Namensgebung ein dazugehöriges XML-Schema generiert, welches die Vorgaben des Referenzmodells und des Archetyps vereint. Diese XML-Schemata sind mit den *operational templates* von openEHR [48] vergleichbar, da sie ebenfalls eigenständige in sich

abgeschlossene Beschreibungen sind.

### **3.4.3 Transformation der EHR-Dokumente**

Im vorherigen Abschnitt wurde die Methode zum Erzeugen eines XML-Schemas vorgestellt, welches die *unique particle attribution constraint rule* nicht mehr verletzt. Da das EHR-Dokument noch die Namensgebung des Referenzmodells befolgt, kann es mit dem auf der beschriebenen Namensgebung basierenden XML-Schema noch nicht validiert werden. Dieselbe Namensgebung muss für den Zweck der Validierung auch auf das EHR-Dokument angewendet werden.

Die EHR-Dokumente werden daher mittels XSLT-Transformationen in EHR-Dokumente mit derselben Namensgebung wie das XML-Schema umgewandelt. Für die ISO/EN 13606 wurde ein generisches XSLT-Skript erzeugt, welches die Archetyp-Knoten-ID an den XML-Element-Namen des jeweiligen XML-Elements anhängt. Das so erzeugte XSLT-Skript ist generisch, umfasst 22 Zeilen Code und kann für jedes ISO/EN-13606-EHR-Extrakt verwendet werden.

In zwei Fällen muss das XSLT-Skript an den Archetyp angepasst werden. Der Fall, dass einzelne Werte auf Elementebene eingeschränkt werden, ist im generischen XSLT-Skript nicht berücksichtigt, kann aber mit XSLT umgesetzt werden, sofern der konkrete Pfad der Werte angegeben wird. Ein weiteres Problem tritt auf, wenn in einem Archetyp eine ungeordnete Liste von Archetyp-Knoten spezifiziert wird, wobei dieselben Knoten mehrfach vorkommen dürfen. In XML-Schema können geordnete Listen die Elemente nur null- oder einmal enthalten. Eine Lösung für dieses Problem kann durch eine künstlich vorgegebene Ordnung im XML-Schema erreicht werden. Im XSLT-Skript werden die jeweiligen Knoten mit einem *xsl:template* sortiert, um dieselbe Reihenfolge wie im XML-Schema vorgegeben zu haben. Der Pfad der XML-Elemente, die sortiert werden sollen, muss im generischen XSLT-Skript angegeben werden.

### **3.4.4 Zusammenfassung der XML-Schema-Validierung**

Der von Rinner et al. präsentierte Ansatz zur semantischen Validierung von EHR-Dokumenten mithilfe von XML-Schema hat den Vorteil, dass die syntaktische Validierung der Vorgaben des Referenzmodells und die semantische Validierung auf derselben Technologie, XML-Schema,

aufbauen. Durch die Verwendung von XML-Schema ist die hierarchische Struktur der Dokumente klar ersichtlich, wodurch die Komplexität des Zwei-Modell-Ansatzes reduziert wird. Weiters existieren für XML-Schema zahlreiche Tools, darunter auch viele Open-Source-Tools und freie Software. Das hier verwendete XML-Schema kann weiters auch für die Transformation von proprietären EHR-Inhalten aus bestehenden Systemen verwendet werden, wie sie in [18] beschrieben wird.



# Entwicklung von Archetypen und archetypkonformen EHR-Dokumenten

In diesem Kapitel werden die im Rahmen des Projektes EHR-Arche erzielten Ergebnisse der iterativen Archetyp-Entwicklung, der Plug-and-Play-Formulargenerierung und der XML-Schema-Validierung vorgestellt. Zusätzlich wird jene Infrastruktur präsentiert, in der die inhaltsbasierte Suche in dezentralen EHR-Systemen mit zentraler Metadatenkomponente implementiert wurde. Die Ergebnisse der inhaltsbasierten Suche selbst werden in Kapitel 5 präsentiert.

Als EHR-Standard wird die ISO/EN 13606 verwendet. Basierend auf ISO/EN-13606-Archetypen werden ISO/EN-13606-EHR-Extrakte erzeugt. Die zugrundeliegende Architektur des EHR-Systems mit dezentraler Datenhaltung setzt auf dem Standard IHE-XDS auf. Die *Document Source* ist im Tool zum Generieren von Formularen aus Archetypen integriert, der *Document Consumer* ist als separate Webapplikation umgesetzt. Für die weiteren Akteure der IHE-XDS-Umgebung werden Komponenten einer kommerziellen IHE-XDS-Lösung genutzt.

## **4.1 Die Entwicklung der Archetypen**

Archetypen sind die Voraussetzung für das Erzeugen von vollstrukturierten ISO/EN-13606-EHR-Extrakten. Um die Zusammenarbeit beim Erstellen der Archetypen zwischen GDAs und Technikern zu unterstützen, wurde die in Kapitel 3.2 vorgestellte Methode zur iterativen Entwicklung von Archetypen verwendet. Im Projekt EHR-Arche wurden die benötigten Archetypen basierend auf Informationsbedürfnissen von Ärzten bei der Behandlung von Diabetes-mellitus-Patienten erstellt. Im Folgenden werden die einzelnen Schritte des iterativen Entwicklungsprozesses und die erzeugten Archetypen beschrieben.

### **4.1.1 Die initiale Planung**

Die im Rahmen des EHR-Arche-Projektes erhobenen Informationsbedürfnisse von Ärzten bei der Behandlung von Diabetes mellitus [145] dienten als Basis für den ersten Schritt des iterativen Entwicklungsprozesses. Um unterschiedliche Aspekte bei der Erhebung der Informationsbedürfnisse festzustellen, wurde die Triangulation [146] verwendet. Die Informationsbedürfnisse wurden mittels Literaturanalyse (Auswertung internationaler diabetesspezifischer Leitlinien), mündlicher Befragung (systematisches Experteninterview in vier unterschiedlichen Krankenhäusern in Tirol), Beobachtung von Patienten beim Arztbesuch (22 Beobachtungen in der Universitätsklinik Innsbruck Tirol) und einer Datenbestandsanalyse in Diabetes-Ambulanzen (Innsbruck, Hall und Wien) erhoben.

Die Informationsbedürfnis-Analyse aus [145] lieferte 471 Informationsbedürfnisse von GDAs während der Behandlung von Diabetes mellitus. Die Informationsbedürfnisse wurden 12 unterschiedlichen Dokumenttypen zugeteilt (Arztbrief, Arztbrief vom Internisten, Arztbrief vom Dermatologen, Arztbrief vom Neurologen, Arztbrief vom Gynäkologen, Arztbrief vom Augenarzt, Arztbrief vom Psychiater, Arztbrief vom Allgemeinmediziner, Patientendokumentation, Konsil, Laborbefund, Medizinische Aufnahme/Anamnese), wobei im Arztbrief alle Informationsbedürfnisse abgebildet wurden, in den anderen Dokumenten jeweils nur ein kontextspezifischer Teil der Informationsbedürfnisse.

Die Informationsbedürfnisse wurden in einer hierarchischen Liste mit sechs Ebenen organisiert. Die erste Ebene besteht aus 11 Kategorien (i.e. Allergianamnese, Selbstmessungen des Patienten, Diabetes-Klassifikation, Allgemeine Anamneseparameter, Psychischer Status, Diabetikerschulung, Diagnosen/Screening, Stattgefundene Operationen,

Durchgeführte Untersuchungen, Laborbefund, Therapie/Medikation), diese enthalten die restlichen 460 Informationsbedürfnisse. Alle Informationsbedürfnisse wurden sowohl in deutscher als auch in englischer Sprache formuliert. Um die Informationsbedürfnisse eindeutig in den Archetypen zu referenzieren, wurde jedem Informationsbedürfnis ein eindeutiger Code zugewiesen. Da Informationsbedürfnisse in unterschiedlichen Kategorien verwendet werden, z.B. "EKG" unter anderem bei "Vitalparameter" als auch bei "Durchgeführte Untersuchungen", wurden insgesamt 404 Codes zugewiesen, die mithilfe der darüberliegenden Kategorien eindeutig den 471 Informationsbedürfnissen zugewiesen werden konnten. Ein Mapping der Archetyp-Knoten auf UMLS, Snomed-CT oder LOINC wird in [147] als sehr zeitaufwendig beschrieben. Da das Mapping nicht im Fokus des EHR-Arche-Projektes lag, wurden die den Informationsbedürfnissen zugewiesenen Codes als Terminologie verwendet. Die so erhaltene Ad-hoc-Terminologie diente als Ersatz für eine standardisierte Terminologie.

Werden idente medizinische Konzepte in mehreren, unterschiedlich strukturierten Archetypen spezifiziert, wird der Grundgedanke der Standardisierung unterwandert und der semantisch interoperable Datenaustausch erschwert. Archetypen können mithilfe von Archetyp-Editoren leicht selber erstellt werden, was wiederum zu einem unkontrollierten Erzeugen von Archetypen führen kann. Um den Wildwuchs von Archetypen zu minimieren wurde versucht, existierende Archetypen wiederzuverwenden. Zu Projektbeginn waren keine ISO/EN-13606-Archetypen öffentlich verfügbar. Aus diesem Grund wurden die openEHR-Archetypen in [65], [148] und [149] für eine Wiederverwendung in Betracht gezogen. Eine Analyse der openEHR-Archetypen ergab, dass 17% der im Projekt EHR-Arche benötigten Informationsbedürfnisse in bestehenden openEHR-Archetypen teilweise abgebildet sind.

OpenEHR-Archetypen können mittels der in [150] präsentierten Methode in ISO/EN-13606-Archetypen umgewandelt werden. Der Konverter ist über ein Webinterface frei zugänglich. Durch eine Umwandlung von openEHR-Archetypen in ISO/EN-13606-Archetypen kann das Problem des Wildwuchses von Archetypen minimiert werden, da zumindest die Struktur von existierenden Archetypen übernommen wird. Es handelt sich bei den umgewandelten Archetypen jedoch um neue Archetypen, die keine Verbindung zu ihrem Quell-Archetyp mehr haben. Ein Anwenden des in ADL vorhandenen Spezialisierungsmechanismus ist aufgrund der unterschiedlichen zugrundeliegenden Referenzmodelle von Quell- und Ziel-Archetyp nicht möglich. Da für die in EHR-Arche

benötigten Informationsbedürfnisse typischerweise jeweils kleine Ausschnitte von openEHR-Archetypen benötigt und aufgrund der unterschiedlichen Referenzmodelle keine Spezialisierung der openEHR-Archetypen möglich war, wurde entschieden die relevanten Ausschnitte in neu erstellten ISO/EN-13606-Archetypen zu klonen.

#### **4.1.2 Erzeugen der Archetypen**

Die im Rahmen des EHR-Arche-Projekts benötigten Archetypen wurden mithilfe des Archetyp-Editors *LinkEHR* [136] erstellt. Der Archetyp-Editor ist in Java programmiert und wird als Open-Source-Software zur Verfügung gestellt. Zusätzlich werden in *LinkEHR* weitere Funktionen zum Erzeugen von EHR-Dokumenten aus existierenden Daten angeboten, die jedoch in der Open-Source-Version nicht enthalten sind und im Rahmen der vorliegenden Arbeit auch nicht verwendet wurden. Neben ISO/EN-13606-Archetypen ist es auch möglich, openEHR- sowie CDA-Archetypen zu erzeugen. Mit der übersichtlichen Benutzeroberfläche können Archetypen erzeugt, Archetyp-Knoten hinzugefügt und deren Eigenschaften verändert werden. Über einen eingebauten Texteditor können auch direkt Änderungen im ADL vorgenommen werden.

Im Gegensatz zu den CDA-Implementierungsleitfäden, die im Rahmen von ELGA zur Verfügung gestellt werden und einen minimal geforderten Datensatz beschreiben, sollten die im Projekt EHR-Arche entwickelten Archetypen im Bereich Diabetes mellitus einen detaillierten Überblick über einen Patienten ermöglichen. Detaillierte Informationen zu Operationen, Befunde von bildgebenden Verfahren, etc. wurden mit den Archetypen nicht abgebildet.

Die erzeugten Archetypen basieren auf den Informationsbedürfnissen, die im ersten Projektabschnitt des EHR-Arche-Projekts gesammelt wurden. Die erhobenen Informationsbedürfnisse wurden über den *Termbinding*-Mechanismus der Archetypen mit einzelnen Archetyp-Knoten verknüpft. Dies ermöglicht es, unabhängig vom Referenzmodell einzelne Knoten in einem Archetyp zu identifizieren. Durch die strukturellen Vorgaben des Referenzmodells gibt es fünf hierarchische Ebenen, denen die Informationsbedürfnisse zugeordnet werden können, nämlich COMPOSITION, SECTION, ENTRY, CLUSTER, ELEMENT. Die oberste Ebene bilden die COMPOSITION-Knoten. Hier werden allgemeine Informationsbedürfnisse wie zum Beispiel "Arztbrief" abgebildet. Je spezifischer die Informationsbedürfnisse, desto tiefer werden sie in der Hierarchie des Referenzmodells

abgebildet. Ein Beispiel für SECTION ist die “Allergianamnese”, für ENTRY die “Rauchgewohnheiten”, für den CLUSTER die “Blutdruckmessung” und für ein ELEMENT “Typ 1a Diabetes”. Zusätzlich ist es möglich, dass Informationsbedürfnisse sich direkt auf Ausprägungen von ELEMENT-Knoten beziehen wie zum Beispiel das Informationsbedürfnis “HbA1C”, welches unter dem ELEMENT “Analyse” (siehe Listing 2.1 auf Seite 19) abgebildet ist. Die unterschiedlichen Ausprägungen von ELEMENTs wurden mittels Archetyp-Knoten-IDs referenziert. Dies erlaubt es, Übersetzungen der Ausprägungen bereitzustellen und den jeweiligen Archetyp-Knoten-IDs *Termbindings* zuzuweisen.

Als Erstes wurden die SECTION-Archetypen erzeugt und mit ENTRY-, CLUSTER- und ELEMENT-Archetypen über *Slots* befüllt. Die Unterbäume der SECTIONS, die in unterschiedlichen COMPOSITION-Archetypen verwendet wurden, variieren. Im openEHR-Modell können mittels Templates Archetypen speziell an einen konkreten Anwendungsfall angepasst werden, indem die nicht benötigten Abschnitte in Archetypen zusätzlich eingeschränkt werden. Im Standard ISO/EN 13606 ist kein Template-Mechanismus vorgesehen, der es erlaubt, die Knoten in einem Archetypen zusätzlich einzuschränken. Die ADL ist prinzipiell in der Lage, über den Spezialisierungsmechanismus Archetypen weiter einzuschränken; für alle in dem spezialisierten Archetyp über *Slots* eingebundenen Archetypen müssen ebenfalls Spezialisierungen erstellt werden, sofern auch diese eingeschränkt werden sollen. Um diese Problematik zu umgehen, wurden für SECTION-Archetypen, bei denen die Unterbäume variieren, nicht die SECTION-Archetypen als *SLOT* eingebunden, sondern deren Inhalte inklusive *Termbindings* etc. im COMPOSITION-Archetyp dupliziert. Dadurch mussten Archetypen, die nur einmal verwendet wurden, nicht spezialisiert werden. Diese Methode wurde nur für SECTIONS angewendet, die direkt im COMPOSITION-Archetyp referenziert werden, alle anderen Archetypen wurden mit dem Spezialisierungsmechanismus umgesetzt.

Existierende medizinische Konzepte wurden in unterschiedlichen Dokumenttypen wiederverwendet. Die zwölf in der Informationsbedarfsanalyse definierten Dokumenttypen wurden als COMPOSITION-Archetypen dargestellt. Der COMPOSITION-Archetyp “Arztbrief” beinhaltet alle erzeugten Informationsbedürfnisse und diente als Ausgangspunkt für alle weiteren COMPOSITION-Archetypen.

Um Archetypen möglichst generisch umzusetzen, wurden etwa im Labor-Archetyp Spezialisierungen verwendet und Werte abhängig von den Geschwisterknoten gesetzt statt über

die hierarchische Struktur. In Listing 2.1 auf Seite 19 wird der ENTRY-Archetyp “Glukose Status” angezeigt, der unter anderem den HbA1c-Wert beschreibt. Der ENTRY-Archetyp “lab\_test” wird in Zeile 4 spezialisiert. Die ELEMENT-Knoten für *Analyse*, *Ergebnis*, *Referenzbereich* und *Interpretation* sind in einem CLUSTER-Archetypen zusammengefasst (Zeile 8–19). Der ELEMENT-Archetyp für die *Analyse* (Zeile 10) wird in dieser Spezialisierung zusätzlich eingeschränkt. Im *Analyse*-Knoten sind die unterschiedlichen Analysen, unter anderem HbA1c, als Ausprägung eingeschränkt (Zeile 13) und können ausgewählt werden. Die *Ergebnisse* (Zeile 15), *Referenzbereiche* (Zeile 17) etc. sind in den Geschwisterknoten abgelegt. Um einen HbA1c-Wert auszuwählen, müssen alle *Analyse*-Knoten mit Ausprägung HbA1c gefunden werden, der eigentliche Wert steht dann aber im Geschwisterknoten *Ergebnis*.

Sind einzelne Archetyp-Knoten eines spezialisierten Archetyps zusätzlich eingeschränkt, wurde die Archetyp-Knoten-ID mit einem Punkt und einer fortlaufenden Nummer gekennzeichnet. In Listing 2.1 in Zeile 10 wurde die Archetyp-Knoten-ID *at0008* mit “.1” erweitert.

Die erzeugten Archetypen besitzen unterschiedliche Granularitäten, d. h. sie beschreiben einzelne medizinische Werte oder ganze Dokumenttypen. In einem Archetyp der Granularität Dokumenttyp wurden unterschiedliche medizinische Konzepte dokumentiert, die in unterschiedlichen Archetypen beschrieben sind. Diese Archetypen wurden mittels *Slots* in anderen Archetypen eingebunden. Der in Abbildung 4.1 abgebildete Arztbrief-Archetyp hat die Granularität Dokumenttyp und umschließt über hundert unterschiedliche Archetypen, die unterschiedliche medizinische Konzepte wie “Laborbefund”, “stattgefundene Operationen” oder “Medikationen” beschreiben. Der Arztbrief-Archetyp ist durch einen Stern in der Mitte der Abbildung zu erkennen.

In Abbildung 4.1 wird nur die Hierarchie der über *Slots* eingebundenen Archetypen gezeigt. Die zusätzlich benötigten Informationen über optionale und obligatorische Teile eines EHR-Dokuments sowie deren Kardinalitäten sind nicht ersichtlich.

Archetypen bieten einen Mechanismus, um die textuellen Informationen in mehreren Sprachen zur Verfügung zu stellen. In den erzeugten Archetypen wurden daher in der *ontology*-Sektion alle Beschreibungen der Archetyp-Knoten-IDs sowohl in deutscher als auch englischer Sprache dargestellt.

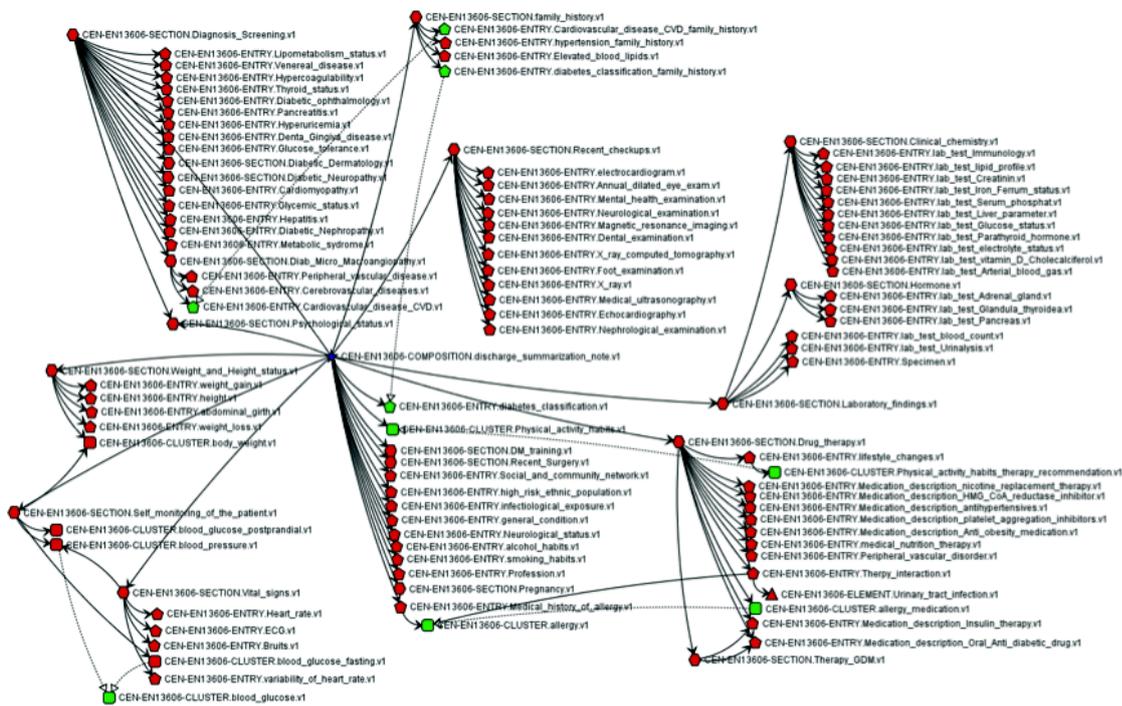


Abbildung 4.1: Der Arztbrief-Archetyp mit allen eingebundenen Archetypen.

Um Knoten innerhalb eines Archetyps zu referenzieren, wurden interne Verweise verwendet. *Assertions*, zur Darstellung von Abhängigkeiten zwischen Archetyp-Knoten (z. B. systolischer Wert muss größer als diastolischer Wert sein etc.), wurden in den erzeugten Archetypen nicht verwendet.

In Abbildung 4.2 werden die im Archetyp “Familienanamnese” verwendeten *Slots* und Spezialisierungen gezeigt. Dieser Archetyp ist in vier COMPOSITION-Archetypen über *Slots* eingebunden und beinhaltet selber vier ENTRY-Archetypen (schwarze Linie mit schwarzem Pfeil). Zwei der eingebundenen Archetypen sind Spezialisierungen von anderen ENTRY-Archetypen (gestrichelte Linie mit weißem Pfeil)

### 4.1.3 Evaluierung und Freigabe der Archetypen

Die Evaluierung der Archetypen erfolgte mit dem in Kapitel 4.2 vorgestellten Tool zum automatischen Generieren von Formularen aus Archetypen. Die Archetypen wurden von einem Techniker und einer Medizinerin kooperativ entwickelt und von der Medizinerin mithilfe der Formulare und von ihr erzeugten Testdokumente getestet.



Abbildung 4.2: Slots und Spezialisierungen im Archetyp “Familienanamnese”.

Die erzeugten Archetypen wurden im *Archetype Repository*, welches in Kapitel 5.1 vorgestellt wird, den GDAs zur Verfügung gestellt. Die 133 erzeugten Archetypen wurden wie folgt den unterschiedlichen Hierarchie-Ebenen des Referenzmodells zugewiesen:

- 12 COMPOSITION-Archetypen
- 18 SECTION-Archetypen
- 87 ENTRY-Archetypen
- 10 CLUSTER-Archetypen
- 6 ELEMENT-Archetypen

34 Archetypen stellten Spezialisierungen anderer Archetypen dar. Rund 380 Mal wurde der *Slot*-Mechanismus verwendet, um Archetypen einzubinden. *Termbindings* wurden rund 1.130 Archetyp-Knoten zugewiesen. Alle Archetypen zusammen enthalten rund 1.083 unterschiedliche Archetyp-Knoten (i.e. 494 ELEMENT-, 299 ENTRY-, 171 SECTION, 107 CLUSTER- und 12 COMPOSITION-Knoten)

## 4.2 Erstellen der archetypkonformen EHR-Dokumente

Für die automatische Generierung von Formularen aus ISO/EN-13606-Archetypen als Unterstützung für das Erzeugen der Testdokumente wurde das im Rahmen des Projekts EHR-Arche entwickelte System *ZK-Arche* [91] verwendet, da keines der analysierten Systeme (siehe Kapitel 2.1.3.1) eine Einbindung in eine IHE-XDS-Umgebung unterstützt.

### 4.2.1 Das System ZK-Arche

Ziel des Systems ZK-Arche ist es, den Benutzer zu unterstützen archetypkonforme EHR-Extrakte für einen Patienten zu erstellen und diese in ein IHE-XDS-*Document Repository* hochzuladen. Die erzeugten EHR-Extrakte wurden als XML-Instanzen lokal gespeichert und zusätzlich in eine IHE-XDS-Umgebung hochgeladen. Das System ZK-Arche verwendet die in Kapitel 3.3 beschriebene Aufteilung in *Model*, *View* und *Controller*.

ZK-Arche wurde auf Basis des Open-Source-AJAX-Webapplikations-Framework ZK [151] entwickelt. Webapplikationen haben unter anderem den Vorteil, dass sie betriebssystemunabhängig sind und neue Softwareversionen für alle Benutzer sofort verfügbar sind. Der Umstand, dass die Webapplikation nur online verwendet werden kann, ist insofern unproblematisch, als für ein Herunterladen der Dokumente aus dem *Document Repository* ebenfalls eine Internetverbindung erforderlich ist. Das ZK-Framework ist eine Server-Plattform, d.h. die Programmlogik wird am Server ausgeführt. Es werden keine speziellen Plug-ins für den Browser benötigt. Eine Webapplikation kann somit auch auf mobilen Endgeräten wie z.B. Tablets (wurde mit Android getestet) verwendet werden.

Die Dateneingabe in ZK-Arche wurde für den Firefox 14 Webbrowser optimiert. Die Aufteilung der unterschiedlichen Komponenten des Userinterface wurde jedoch so gewählt, dass Formulare auch auf mobilen Endgeräten verwendbar sind. Für die Darstellung der Formularkomponenten wurde im Gegensatz zu [56] auf aufklappbare Komponenten und nicht auf Overlays gesetzt, um sich wiederholende Knoten in Archetypen darzustellen. In Abbildung 4.3 ist die Aufteilung des Userinterface ersichtlich. Im Kopfbereich befindet sich das Logo und der Logout-Button sowie ein Menü zum Administrieren der Patienten, Ärzte und Archetypen. Um eine realistische IHE-XDS-Umgebung abzubilden, ist es möglich, unterschiedliche Ärzte, mit einer unterschiedlichen Spezialisierung, als Autoren auszuwählen. Das Hauptfenster ist in einen linken und einen rechten Teil unterteilt. Links können Patienten und die dazugehörigen Dokumente ausgewählt werden. Standardmäßig wird das Dokument nach der Auswahl im rechten Fenster als HTML-Dokument angezeigt. Neben der HTML-Ansicht kann noch eine XML-Ansicht und die Editier-Ansicht (siehe Abbildung 4.4) gewählt werden.

Die HTML-Ansicht (siehe Abbildung 4.3) ermöglicht es, den Inhalt des EHR-Extrakts in menschenlesbarer Form wiederzugeben. Ein generisches XSLT-Skript wurde auf ein



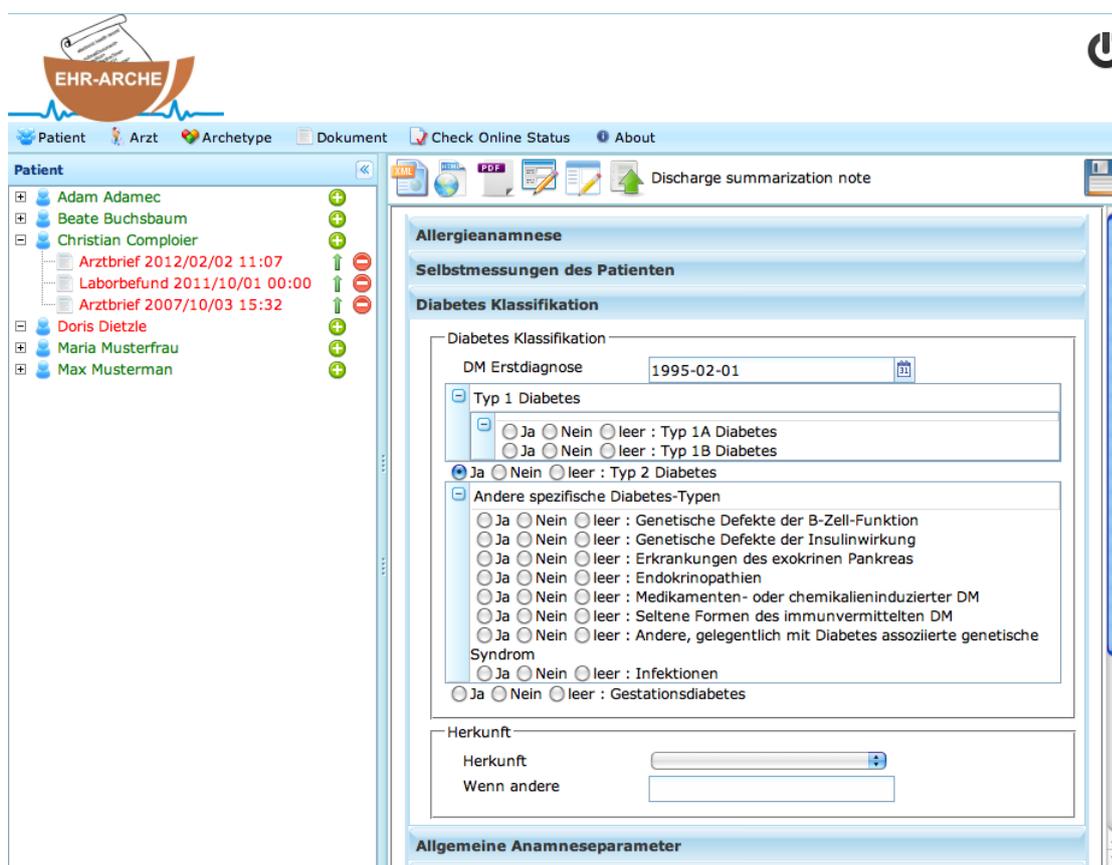
Abbildung 4.3: HTML-Darstellung der EHR-Dokumente in ZK-Arche.

EHR-Extrakt angewendet und transformiert dieses in ein HTML-Dokument. Das XSLT-Skript wurde von [136] übernommen und adaptiert. Es wurde unter anderem eine deutsche Übersetzung zum XSLT-Skript hinzugefügt sowie die Font-Größe, die Darstellung der CLUSTER-Klasse und des Kopfbereichs sowie die Darstellung der Zeitformate geändert.

Zu jedem ISO/EN-13606-EHR-Extrakt kann ein PDF-Dokument hochgeladen werden. Die PDF-Dokumente wurden für die Evaluierung verwendet, um die herkömmliche Suche basierend auf Metadaten und die neu entwickelte inhaltsbasierte Suche in ISO/EN-13606-EHR-Extrakten miteinander zu vergleichen.

## 4.2.2 Formulargenerierung aus Archetypen

In Abbildung 4.4 ist ein Ausschnitt aus dem Editiermodus des ZK-Arche-Systems zu sehen. Wird ein Dokument editiert, werden basierend auf dem COMPOSITION-Archetyp, der dem Dokument zugrundeliegt, alle Archetypen zu einem *comprehensive Archetype* [136] zusammengefasst, der alle eingebundenen Archetypen sowie die Knoten des Referenzmodells beinhaltet. Dies ist nötig, da ein Archetyp nur jene Knoten des Referenzmodells beinhaltet, die zusätzlich eingeschränkt sind. Der *comprehensive Archetype* wird für die Formulargenerierung abgearbeitet und dabei für jeden Knoten die für die jeweilige Klasse des Referenzmodells festgelegte Visualisierung geladen.



The screenshot displays the ZK-Arche system interface. On the left, a patient list includes Adam Adamec, Beate Buchsbaum, Christian Comploier, Doris Dietzle, Maria Musterfrau, and Max Musterman. The main area shows a form for 'Discharge summarization note' with sections for 'Allergianamnese', 'Selbstmessungen des Patienten', and 'Diabetes Klassifikation'. The 'Diabetes Klassifikation' section is expanded, showing 'DM Erstdiagnose' as '1995-02-01' and several sub-sections with radio button options for 'Ja', 'Nein', and 'leer'.

Abbildung 4.4: Beispiel eines Arztbrief-Formulars in ZK-Arche.

Die einzelnen Klassen des Referenzmodells wurden im Formular zur Dateneingabe unterschiedlich visualisiert. Der hierarchische Aufbau der Klassen wurde im Formular berücksichtigt, siehe Abbildung 4.4. COMPOSITIONs wurden als Formular dargestellt.

SECTIONs sind Formularblätter und wurden in Form von *Accordions* (in ZK als *Tab-Box* bezeichnet), die beim Auswählen jeweils den ganzen Bildschirm ausfüllen, dargestellt. ENTRYs wurden durch einen Rahmen, der mit dem Namen des jeweiligen ENTRYs beschriftet ist, voneinander getrennt (in ZK als *GroupBox* bezeichnet). Für CLUSTER wurden zusammenklappbare Boxen verwendet, ELEMENTs wurden direkt in darüber liegenden Komponenten angezeigt. Abhängig von den Datentypen wurden Textfelder, Radiobuttons, Auswahllisten oder ein Widget zum Auswählen eines Datums bzw. eines Zeitpunktes angeboten. Kann eine Klasse öfter vorkommen, wurde im Kopfbereich der jeweiligen Visualisierung ein grünes Plus angezeigt, mit dessen Hilfe die jeweilige Komponente dupliziert werden kann. Schreibt das *Model* für einen Wert eine einzelne Ausprägung vor, so kann das ELEMENT im Formular nicht editiert werden, sondern wurde im Hintergrund mit dieser Ausprägung belegt.

Ausgehend von einem ISO/EN-13606-COMPOSITION-Archetyp wurde ein *comprehensive Archetype* erstellt, als Instanz des erweiterten AOM (siehe Kapitel 3.3) persistiert und als *Model* verwendet. Dieser diente zur Erzeugung der Formularkomponenten. Wird durch den Archetyp eine Mehrfachdokumentation in einer Formularkomponente erlaubt, so kann diese dupliziert werden. Dazu wurde im *comprehensive Archetype* die jeweilige Komponente geklont und hinzugefügt. Es existierte zu jeder Eingabe im Formular eine idente Repräsentation als Instanz des *comprehensive Archetypes*. Beim Speichern wurde aus diesem *comprehensive Archetype* direkt das EHR-Extrakt generiert.

### **4.2.3 Erzeugen des *comprehensive Archetype***

Der *comprehensive Archetype* für die Erzeugung eines Dokuments muss die Granularität EHR\_EXTRACT haben. Ein Dokument entspricht zwar einer COMPOSITION, es werden aber Metainformationen wie etwa die Patienten-ID aus der EHR\_EXTRACT-Klasse benötigt. Im ersten Schritt wurden die Klassen, die auf die EHR\_EXTRACT-Klasse fehlen, zum Archetyp hinzugefügt. Zu einem CLUSTER-Archetyp wurden beispielsweise ein ENTRY, eine COMPOSITION und die EHR\_EXTRACT-Klasse hinzugefügt. Der *comprehensive Archetype* erhielt somit auch eine andere ID, die mit CEN-EN13606-EHR\_EXTRACT beginnt.

Im nächsten Schritt wurden die obligatorischen Attribute des Referenzmodells, die im Archetyp nicht eingeschränkt sind, hinzugefügt. Die optionalen Attribute wurden nicht

angegeben, da der *comprehensive Archetype* sonst sehr groß geworden wäre. Es kann auch davon ausgegangen werden, dass, wenn optionale Klassen und Attribute des Referenzmodells im *comprehensive Archetype* benötigt werden, diese im Archetyp als verpflichtend gekennzeichnet sind und somit im *comprehensive Archetype* wieder aufscheinen. Werte, die vom Referenzmodell vorgegeben sind wie zum Beispiel gewisse Codesysteme, wurden direkt im *comprehensive Archetype* hinzugefügt.

Alle Archetypen, die mittels *Slots* referenziert sind, wurden rekursiv direkt im Archetyp eingefügt. Eine Limitation des Archetype Object Model 1.4 liegt darin, dass die *ontology*-Sektion für alle Archetypen zentral angegeben ist. Die Archetyp-Knoten-ID *at0001* kann etwa nur einmal beschrieben, übersetzt oder mit einer Terminologie verbunden werden. Um dieselben Archetyp-Knoten-IDs eines eingebundenen Archetyps abzubilden und die Eindeutigkeit der Archetyp-Knoten-IDs über Archetypen hinweg sicherzustellen, wurden diese in Form einer Punktnotation mit der ID der eingefügten ATs erweitert.

Der Knoten "Allergianamnese" mit der Archetyp-Knoten-ID *at0063* im Archetyp CEN-EN13606-ENTRY.Medical\_history\_of\_allergy.v1 wurde beispielsweise im *comprehensive Archetype* zu *at067069078045069078049051054048054045069078084082089.07710110010509909710809510410511511611114121095111102095097108108101114103121.118049.0063*. Jedes Zeichen und jeder Buchstaben der Archetyp-ID wurde mit dem jeweiligen ASCII-Code ersetzt, um eine gültige Archetyp-Knoten-ID zu bekommen.

#### **4.2.4 Die Testdokumente**

Mithilfe des ZK-Arche-Systems wurden zwei Patienten angelegt. Für Testpatient A wurden im Zeitbereich von November 2001 bis September 2011 insgesamt 53 Dokumente jeweils als PDF und als ISO/EN-13606-EHR-Extrakt hochgeladen, für Testpatient B im Zeitbereich von März 2009 bis Oktober 2011 jeweils 29 PDF-Dokumente und 29 ISO/EN-13606-EHR-Extrakte. Die ISO/EN-13606-EHR-Extrakte wurden mithilfe des ZK-Arche-Systems von einer Medizinerin erzeugt. Die darauf basierenden PDF-Dokumente wurden ebenfalls von der Medizinerin in Word erstellt und zu den ISO/EN-13606-EHR-Extrakten im ZK-Arche-System hinzugefügt.

Die so erzeugten ISO/EN-13606-EHR-Extrakte umfassen:

- 82 COMPOSITIONs
- 531 SECTIONs
- 1804 ENTRYs
- 2798 CLUSTERs
- 10728 ELEMENTs

Die 82 XML-Dokumente zusammen haben eine Größe von 19,2 MB, die 82 PDF-Dokumente 6,47 MB.

Die Dokumente wurden für die Evaluierung im Rahmen des EHR-Arche-Projekts verwendet (siehe Kapitel 5.6). Um eine sinnvolle Evaluierung mit Ärzten und Ärztinnen durchführen zu können, mussten die Dokumente auf realistischen Daten beruhen, wobei die unterschiedlichen Werte in den unterschiedlichen Dokumenten eine konsistente Krankengeschichte darstellen mussten.

### 4.3 Validierung der Testdokumente

Für die automatische Generierung des benötigten XML-Schemas aus ISO/EN-13606 Archetypen (siehe Kapitel 3.4) wurden zwei unabhängige Implementierungen umgesetzt.

Die erste Implementierung wurde in [141] vorgestellt. Das in Java programmierte Tool geht von einem ISO/EN-13606-EHR-Extrakt aus und erkennt die darin verwendeten Archetypen anhand der referenzierten Archetyp-IDs. Mithilfe der Archetyp-IDs wurden die entsprechenden Archetypen geladen und zusammen mit dem Referenzmodell zu einem *comprehensive Archetype* zusammengestellt. Aus dem *comprehensive Archetype* wurde ein XML-Schema generiert. In diesem XML-Schema wurden die Archetyp-Knoten-IDs verwendet, um XML-Schema-Elemente eindeutig zu benennen. Das EHR-Extrakt wurde mittels des generischen XSLT-Skripts transformiert und das erzeugte XML-Schema angewendet. Die Fehler wurden mit Angabe der Zeilennummer am Bildschirm ausgegeben.

Für das Erzeugen des XML-Schemas wurde in der zweiten Implementierung der *Comprehensive Archetype*, der zum Erzeugen der Formulare in der Plug-and-Play-Formulargenerierung im System ZK-Arche erzeugt wird, verwendet.

Der auf XML-Schema basierende Ansatz zur Validierung von EHR-Dokumenten wurde auch an CDA-Dokumenten und ELGA-Laborbefunden getestet [152].

## 4.4 IHE-XDS-Testumgebung

Im Folgenden wird die Implementierung der IHE-XDS-Umgebung beschrieben, die in weiterer Folge als Basis für die Umsetzung der inhaltsbasierten Suche in Kapitel 5 herangezogen wurde.

Als IHE-XDS-Umgebung wurde die im Rahmen des EHR-Arche-Projekts verwendete kommerzielle Lösung *SENSE* [92] von ITH-icoserve verwendet. Die Firma ITH-icoserve und der Projektpartner UMIT waren maßgeblich an der Entwicklung des health@net-Projekts [153] beteiligt. Das health@net-Projekt lieferte eine 2009 fertiggestellte Open-Source-IHE-XDS-Implementierung, die als Ausgangspunkt der *SENSE*-Implementierung gesehen werden kann. *SENSE* bildet die Basis für das Gesundheitsnetz Tirol (GNT) und bietet sogenannte *Document Consumer Adapter* (Abbildung 4.5), die es erlauben, über spezielle Programmbibliotheken ITI-15-, ITI-17- und ITI-18-Transaktionen zu einem *Document Repository* und einer *Document Registry* durchzuführen. Die Standard-Akteure wie *Patient Identity Source*, *Document Repository* und *Document Registry* wurden von SENSE gehostet.

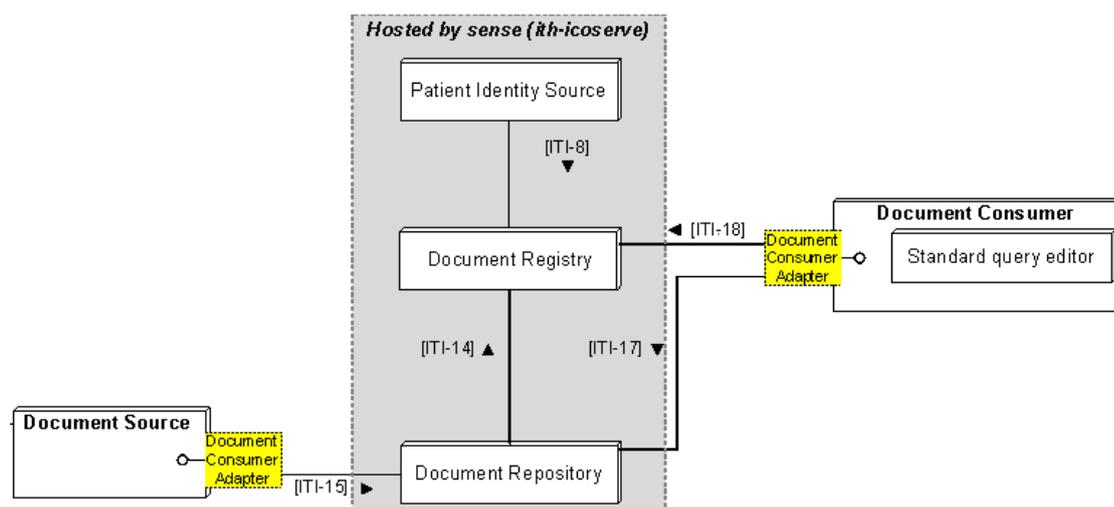


Abbildung 4.5: Akteure in der IHE-XDS-Umgebung SENSE.

Mithilfe des *Document Consumer Adapters* wurde im Rahmen des Projektes EHR-Arche ein

standardkonformer *Document Consumer*-Akteur entwickelt. Hauptverantwortlich für die Implementierung des *Document Consumers* war der Projektpartner UMIT. Über Webservice-Aufrufe konnten dabei vom *Document Consumer* Suchanfragen als ITI-18-Transaktionen an das *Document Registry* gesendet werden. Über ITI-17-Transaktionen wurden die Dokumente vom *Document Repository* heruntergeladen. Als *Document Source* diente das ZK-Arche-System (siehe Kapitel 4.2), welches über die ITI-15-Transaktion und den *Document Consumer Adapter* Dokumente in die IHE-XDS-Umgebung hochladete.

Der *Document Consumer* wurde als Webapplikation realisiert und mit dem Open-Source-AJAX-Webapplikations-Framework ZK [151] umgesetzt. Nach einem Log-in wird die Weboberfläche wie in Abbildung 4.6 dargestellt geladen. In der Menü-Leiste werden administrative Funktionen zur Verfügung gestellt. Über das Menü *Modus* kann zwischen den Modi *Standard* und *Erweitert* gewählt werden. Der Modus *Standard* entspricht einer herkömmlichen Suche in IHE-XDS, wie sie in Abbildung 4.6 abgebildet ist und in diesem Kapitel erläutert wird. Mit der Funktion *Erweitert* wird die im Rahmen der vorliegenden Arbeit entwickelte inhaltsbasierte Suchanfrage im *Document Consumer* aufgerufen, die in Kapitel 5 im Detail beschrieben wird.

Im Modus *Standard* können nach Auswahl eines Testpatienten drei unterschiedliche Metadaten zum Filtern der Dokumente eingeschränkt werden. Es können Dokumente aus dem gesamten Zeitbereich oder aus einem frei definierten Zeitbereich ausgewählt werden. Die Auswahlmöglichkeiten sind im Screenshot ersichtlich. Eine weitere Auswahlmöglichkeit steht für die Dokumenttypen zur Verfügung, wobei fünf unterschiedliche Dokumenttypen unterstützt werden (i.e. Arztbriefe, Laborbefunde, Medizinische Aufnahme, Konsil und Patientendokumentation). Über Checkboxes können ein oder mehrere Dokumenttypen ausgewählt werden. Ist kein Dokumenttyp ausgewählt, wird automatisch nach allen Dokumenttypen gesucht. Als letzte Auswahlmöglichkeit kann die Fachrichtung des Arztes, der das jeweilige Dokument erstellt hat, eingeschränkt werden. Dies ist in der vorliegenden Arbeit hauptsächlich für die Einschränkung der Arztbriefe interessant, da diese im Gegensatz zu den anderen Dokumenttypen von unterschiedlichen Fachrichtungen erstellt werden (i.e. Arztbrief vom Internisten, Arztbrief vom Dermatologen, Arztbrief vom Neurologen, Arztbrief vom Gynäkologen, Arztbrief vom Augenarzt, Arztbrief vom Psychiater, Arztbrief vom Allgemeinmediziner).

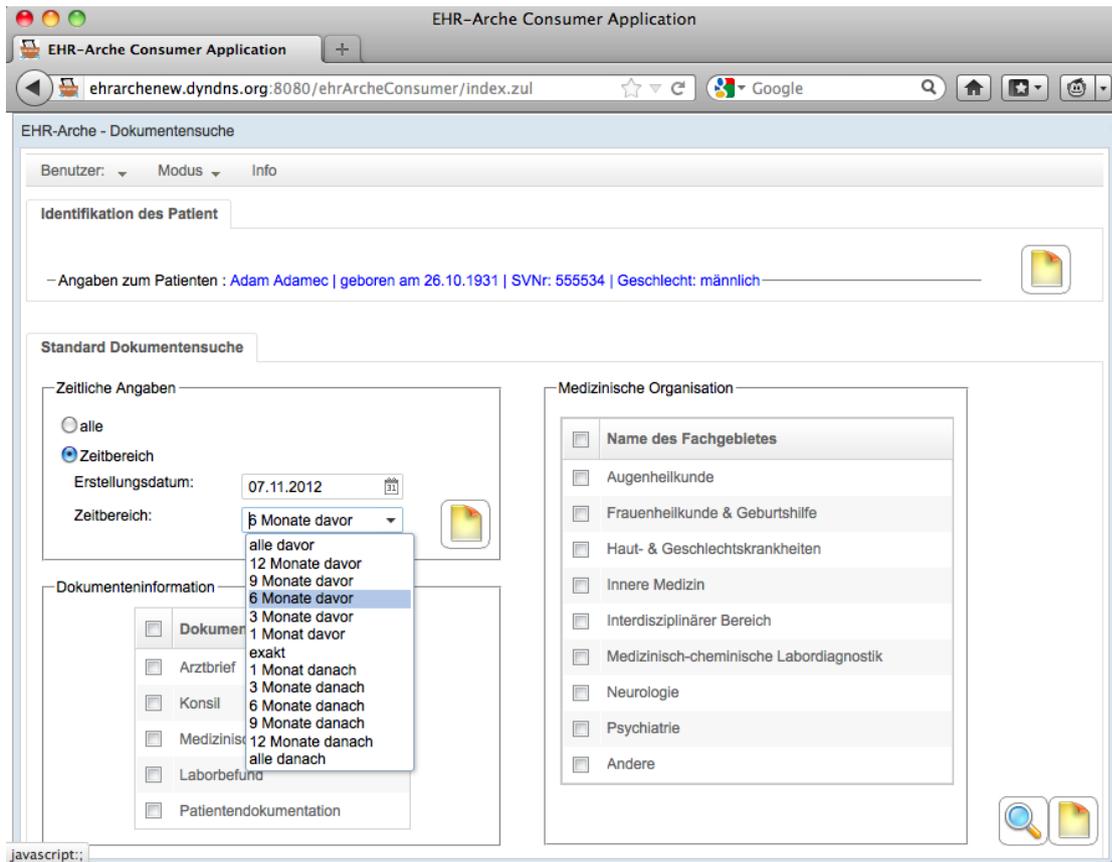


Abbildung 4.6: Umsetzung der Suchoberfläche im *Document Consumer*.

Das Userinterface für die metadatenbasierte Suche wurde 2010 entwickelt und entspricht in großen Teilen dem 2012 präsentierten Konzept für das Patientenportal in ELGA [9]. Es erlaubt ebenfalls die Einschränkung des Datums, der erstellenden Organisation, der Fachrichtung sowie der Dokumenttypen. Die Einschränkungen werden im ELGA-Patientenportal im Gegensatz zum in der vorliegenden Arbeit vorgestellten Konzept als Filter bezeichnet, die Suchergebnisse werden direkt neben dem Filter dargestellt und nicht in einem zusätzlichen Tab wie im EHR-Arche-Projekt.

Mit der Lupe im unteren rechten Bereich wird eine ITI-18-Transaktion mit den ausgewählten Parametern vom *Document Consumer* an die *Document Registry* abgesendet. Das Ergebnis wird in einem zweiten Tab *Präsentation der Ergebnisse* (siehe Abbildung 2.8) angezeigt. Die zurückgelieferten Dokumente werden in chronologischer Reihenfolge angezeigt. Es werden jeweils sieben Dokumente angezeigt, weitere Dokumente können über Navigationstasten am

unteren Bildrand angezeigt werden. Um eine neue Suche auszuführen, muss wieder auf den Tab *Standard Dokumentsuche* geschaltet werden.

Diese Darstellung der Ergebnisse wird auch im Rahmen des Patientenportals, welches im Rahmen von ELGA umgesetzt wird, gewählt [9]. In ELGA werden zusätzlich Spalten für die erstellende Organisation, die Fachrichtung sowie den Dokumenttyp angegeben. Da im Patientenportal auch eine Rechteverwaltung integriert ist, können auch mehrere Dokumente gleichzeitig ausgewählt werden. Außerdem werden Informationen zum Sperrstatus gegeben.

Die *Document Source* wurde im Projekt EHR-Arche direkt in das ZK-Arche-System integriert. Neben der Möglichkeit, einzelne Dokumente hochzuladen, wurde ein Hochladen aller Dokumente zu einem Patienten zur Verfügung gestellt (siehe Abbildung 4.7). Über die Pfeil-Knöpfe werden die jeweiligen Dokumente eines Patienten in das *Document Repository* hochgeladen. Es wird dabei zwischen dem Hochladen der ISO/EN-13606-EHR-Extrakte als strukturierte bzw. der PDF-Dokumente als unstrukturierte Dokumente unterschieden.

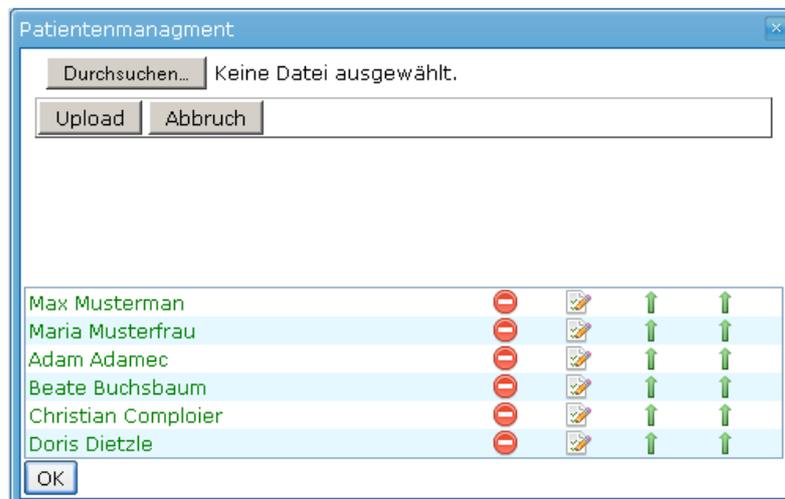


Abbildung 4.7: Dialogbox der *Document Source* um Dokumente ins *Document Repository* hochzuladen.

Jedem Patient wurden jeweils unstrukturierte und inhaltlich idente strukturierte Dokumente zugeordnet. Die unstrukturierten Dokumente basierten auf dem *mimetype* *application/octet-stream* sowie den *format-code* *ScanPDF/IHE 1.x*, für die strukturierten Dokumente wurde der *mimetype* *text/xml* sowie der *format-code* *EN13606* verwendet.

## Inhaltsbasierte Suche in IHE-XDS

In diesem Kapitel wird die Anwendung der Methoden, die allgemein für EHR-Dokumente und eine inhaltliche Suche bei dezentraler Datenhaltung mit zentraler Metadatenkomponente vorgestellt wurden, auf ISO/EN-13606-Archetypen, ISO/EN-13606-EHR-Extrakte und eine inhaltliche Suche in einer IHE-XDS-Umgebung im Rahmen des Projekts EHR-Arche beschrieben.

Die Suche in einer IHE-XDS-Umgebung wurde wie im Folgenden beschrieben umgesetzt. Als Wissensbasis wurde ein *Archetype Repository* eingeführt. Als zentrale Metadatenkomponente wurde die *Document Registry* verwendet. Die EHR-Dokumente vom GDA wurden mittels der *Document Source* in das *Document Repository* abgelegt. Das Ausführen der Suche und Aufbereiten der Suchergebnisse wurde in den neu eingeführten *Document Crawler* ausgelagert, um existierende *Document Consumer* bei den GDAs mit möglichst wenig Entwicklungsaufwand einbinden zu können. In Abbildung 5.1 werden die neuen Akteure *Document Crawler* und *Archetype Repository* sowie der erweiterte *Document Consumer* mit deren Verbindungen zu den bestehenden Akteuren dargestellt.

In Abbildung 5.2 sind die unterschiedlichen Interaktionen zwischen den Akteuren im Zuge einer inhaltsbasierten Suche abgebildet. Der *Document Consumer* lädt beim Starten die Liste der möglichen Suchbegriffe vom *Archetype Repository*. Mithilfe dieser Liste kann der GDA im *Document Consumer* eine inhaltsbasierte Suchanfrage formulieren und an den *Document Crawler* senden. Im *Document Crawler* wird die inhaltsbasierte Suchanfrage in eine

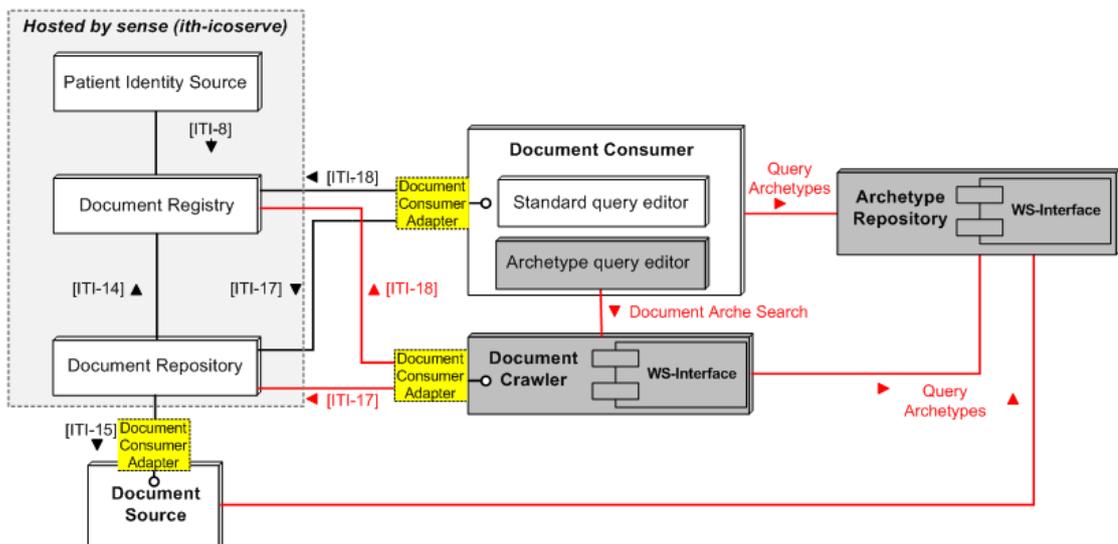


Abbildung 5.1: Überblick IHE-XDS-Architektur mit zusätzlichen Akteuren zur Unterstützung einer inhaltsbasierten Suche. Die zusätzlichen Akteure sind grau hinterlegt. Die Akteure, die von *sense* zur Verfügung gestellt wurden, sind von einem hellgrauen Rahmen umgeben.

Metadaten- sowie eine XQuery-Suchanfrage umgewandelt. Für die metadatenbasierte Suchanfrage werden die relevanten Dokumenttypen aus dem *Archetype Repository* geladen, eine metadatenbasierte Suchanfrage generiert und an die *Document Registry* gesendet. Zum Erzeugen der XQuery werden zu jedem Suchbegriff die jeweiligen Informationen aus dem *Archetype Repository* geladen. Die gefundenen Dokumente der metadatenbasierten Suche werden vom *Document Repository* geladen, die XQuery auf die Dokumente angewendet, die Ergebnistabelle erzeugt und an den *Document Consumer* zurückgesendet.

Im Folgenden werden das *Archetype Repository*, das Erzeugen der inhaltsbasierten Suchanfrage, das Erzeugen der metadatenbasierten Suchanfrage und der XQuery-Suchanfrage aus der inhaltsbasierten Suchanfrage, die Aufbereitung der Ergebnisse sowie die Ergebnisse der Evaluierung beschrieben.

## 5.1 Das ISO/EN-13606-*Archetype Repository*

Als Wissensbasis wurde der neue Akteur *Archetype Repository* eingeführt. Archetypen übernehmen bei der inhaltsbasierten Suche eine zentrale Rolle. Sie werden verwendet, um die EHR-Dokumente in der *Document Source* zu erzeugen und zu validieren, um im *Document*

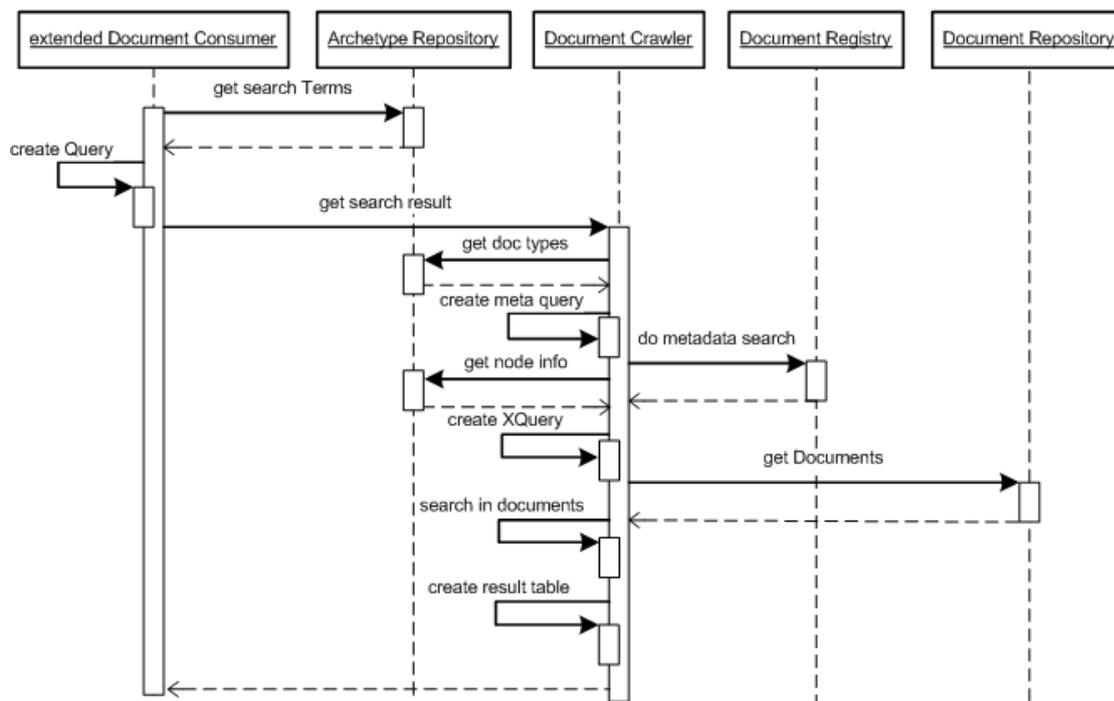


Abbildung 5.2: Überblick über Interaktionen zwischen den Akteuren bei der inhaltsbasierten Suche.

*Consumer* die Suchanfragen zu erstellen, sowie um im *Document Crawler* die EHR-Dokumente zu durchsuchen und die Suchergebnisse aufzubereiten. Der Fokus des erzeugten *Archetype Repository* ist es, die Akteure in einer IHE-XDS-Umgebung bei den jeweiligen Aufgaben zu unterstützen. Im *Archetype Repository* werden die verwendeten Archetypen indiziert, um relevante Informationen, Verknüpfungen, Einschränkungen etc. den beteiligten Akteuren zur Verfügung zu stellen. Es stellt neben den einzelnen Archetypen das nötige Wissen zum Erzeugen der inhaltsbasierten Suchanfrage, der abgeleiteten metadatenbasierten Suchanfrage, der XQuery-Suchanfrage und der Visualisierung der Ergebnisse zur Verfügung.

In Kapitel 3.1.1 sind die Anforderungen an das *Archetype Repository* zusammengefasst. Die Archetyp-Repositorys, die in Kapitel 2.1.3.3 vorgestellt werden, haben einen anderen Fokus und erfüllen die Anforderungen nur zum Teil bzw. stehen nicht als Open-Source-Software zur Verfügung. Daher wurde im Projekt EHR-Arche ein eigenes *Archetype Repository* entwickelt.

### 5.1.1 Datenstruktur des *Archetype Repository*

Das *Archetype Repository* wurde mit Semantic-Web-Technologien über einen *Triple Store* realisiert. Es wurden drei RDF-Dokumente für das *Archetype Repository*, für die Datentypen sowie für die Informationsbedürfnisse definiert. Den drei *Namespaces* ATS, DATATYPE und TERMS wurden folgende URLs zugewiesen.

- *Archetype-Repository*: `xmlns:ATS="http://mias.meduniwien.ac.at/ATS/"`
- Datentypen: `xmlns:DATATYPE="http://mias.meduniwien.ac.at/DATATYPE/"`
- Informationsbedürfnisse: `xmlns:TERMS="http://mias.meduniwien.ac.at/TERMS/"`

Ziel des *Archetype Repository* ist es, durch die verwendeten *RDF-Triple* die Suche über Archetypen zu erleichtern. Das definierte Vokabular wurde speziell auf die Bedürfnisse des Prototypen abgestimmt, es sind jedoch nicht alle Konzepte der Archetypen integriert. Die Archetypen-IDs werden genutzt, um alle Archetypen und deren Knoten eindeutig zu referenzieren und unterscheiden zu können. Da die verwendeten EHR-Dokumente auf dem Standard ISO/EN 13606 basieren, wurden die speziellen Eigenschaften des Standards ISO/EN 13606, wie etwa die Datentypen, im *Archetype Repository* berücksichtigt.

Das Vokabular des *Archetype Repository* wurde im *Namespace* ATS definiert. Eine Übersicht des Vokabulars ist in Abbildung 5.3 ersichtlich. Über *ATS:isSpecializationOf*, *ATS:isSpecializedBy* werden Verknüpfungen zwischen spezialisierten Knoten in unterschiedlichen Archetypen aufgebaut. Über *ATS:parent* und *ATS:child* werden die Eltern-Kind-Beziehungen zwischen den Knoten in einem Archetyp, aber auch zwischen unterschiedlichen Archetypen definiert. Wird ein Archetyp über einen *Slot* in einen anderen Archetyp eingebunden, ist der eingebundene Archetyp über *ATS:child* mit dem *Slot*-Knoten verbunden, der eingebundene Archetyp referenziert über *ATS:parent* den Archetyp, in den er eingebunden ist.

Zu jedem Archetyp-Knoten werden noch weitere Informationen benötigt. Über *ATS:nodeID* ist die Archetyp-Knoten-ID des jeweiligen Archetyp-Knoten ersichtlich, *ATS:labelDe*, *ATS:labelEn* verbinden die Texte in der jeweiligen Sprache aus der *ontology*-Sektion des Archetyps mit den jeweiligen Archetyp-Knoten. Die Verknüpfung zur zugrundeliegenden

Referenzmodellklasse wird in *ATS:rmClass* gemacht. Für Archetyp-Knoten, die elementare Werte beschreiben, also die *ATS:rmClass* ELEMENT haben, sind abhängig vom zugrundeliegenden Datentypen (*ATS:elementDataType*) unterschiedliche Variablen vorhanden. Mehrere Ausprägungen für einen Wert werden mittels *ATS:elementDropDownValue* angegeben, Ausprägungen, die über eine Archetyp-Knoten-ID definiert sind, werden über *ATS:elementDropDownValueCodePhrase* und den zugrundeliegenden Archetyp-Knoten angegeben. Sind diesen Ausprägungen Codes zugewiesen, werden diese über *ATS:codePhraseValue* referenziert. Ausprägungen unterschiedlicher Einheiten beim *Physical Quantity*-Datentyp werden über *ATS:pqUnitValue* angegeben.

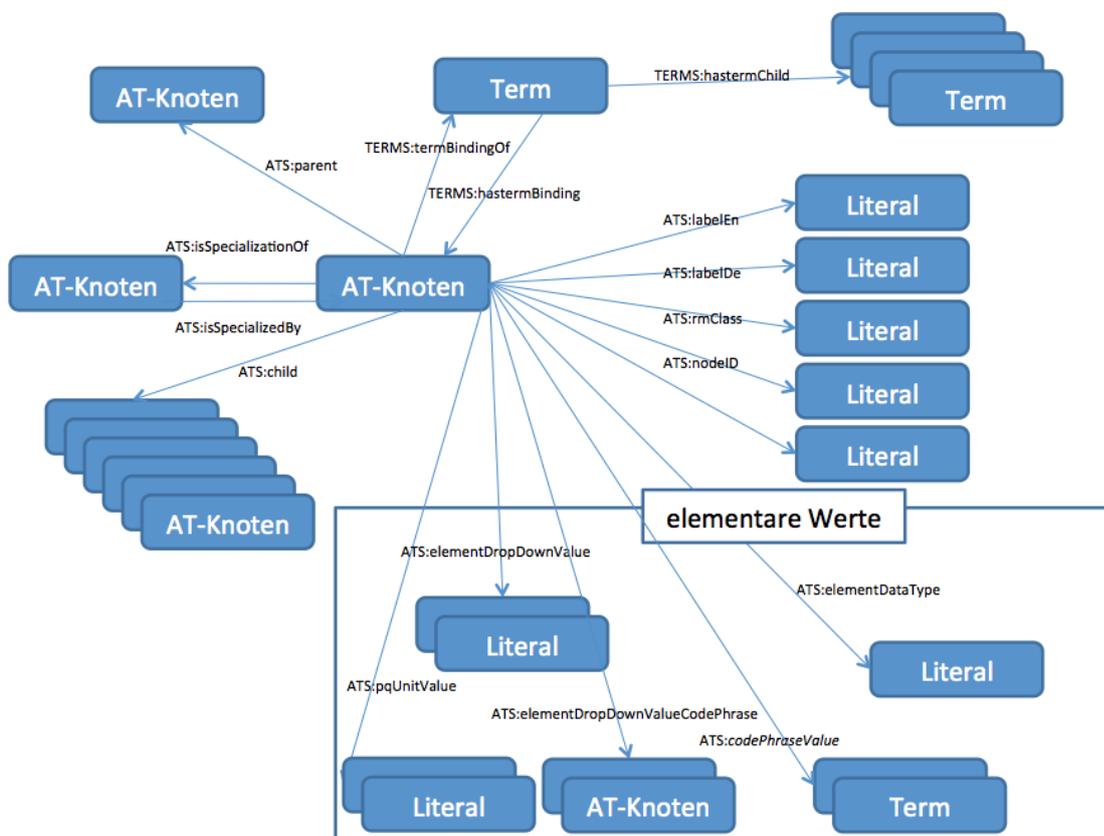


Abbildung 5.3: Überblick über die im Archetype Repository verwendeten RDF-Klassen.

Ein Ausschnitt der *RDF-Triple* ist in Listing 5.1 ersichtlich. Jedem Root-Knoten eines Archetyps wird eine eindeutige URL zugewiesen. Die URL besteht aus der Archetyp-ID und der URL des MIAS Wissenschafts-Webserver (i.e <http://mias.meduniwien.ac.at>). In Zeile 7 ist eine eindeutige

Listing 5.1: Beispiel *RDF-Triple* des *Archetype Repository* in XML

```

1 <rdf:RDF>
2 ...
3 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4 xmlns:TERMS="http://mias.meduniwien.ac.at/TERMS/"
5 xmlns:DATATYPE="http://mias.meduniwien.ac.at/DATATYPE/"
6 xmlns:ATS="http://mias.meduniwien.ac.at/ATS/"
7 <rdf:Description rdf:about="http://mias.meduniwien.ac.at/ATS/CEN-EN13606-
  CLUSTER.allergy_medication.v1/">
8   <ATS:child rdf:resource="http://mias.meduniwien.ac.at/ATS/CEN-EN13606-
  CLUSTER.allergy_medication.v1/parts%3Cat0066.1%3E"/>
9   <TERMS:hasTermBinding rdf:resource="http://mias.meduniwien.ac.at/TERM
  /98"/>
10  <ATS:nodeID xml:lang="de">CEN-EN13606-CLUSTER.allergy_medication.v1/
  at0063.1</ATS:nodeID>
11  <ATS:child rdf:resource="http://mias.meduniwien.ac.at/ATS/CEN-EN13606-
  CLUSTER.allergy_medication.v1/parts%3Cat0084%3E"/>
12  <ATS:iSpecializationOf rdf:resource="http://mias.meduniwien.ac.at/ATS/
  CEN-EN13606-CLUSTER.allergy.v1/">
13  <ATS:childrenSeq rdf:nodeID="A76"/>
14  <ATS:child rdf:resource="http://mias.meduniwien.ac.at/ATS/CEN-EN13606-
  CLUSTER.allergy_medication.v1/parts%3Cat0077%3E"/>
15  <ATS:parent rdf:resource="http://mias.meduniwien.ac.at/ATS/CEN-EN13606-
  SECTION.Drug_therapy.v1/members%3Cat0005%3E"/>
16  <ATS:labelen xml:lang="en">Allergy Medication</ATS:labelen>
17  <ATS:child rdf:resource="http://mias.meduniwien.ac.at/ATS/CEN-EN13606-
  CLUSTER.allergy_medication.v1/parts%3Cat0073%3E"/>
18  <ATS:rmclass xml:lang="en">CLUSTER</ATS:rmclass>
19  <ATS:labelde xml:lang="de">Allergie Medikation</ATS:labelde>
20 </rdf:Description>
21 ...
22 </rdf:RDF>

```

URL ersichtlich. Jeder weitere Knoten im Archetyp erhält eine eindeutige URL, die über den Archetyp-Pfad zusammengesetzt wird. Dabei werden immer `/attr name[object id]/.../attr name[object id]` aneinandergereiht. Da die eckigen Klammern (`[` und `]`) nicht in einer URL vorkommen dürfen, werden diese beiden Zeichen mit `%3C` und `%3E` ersetzt. In Zeile 8 ist ein Beispiel einer URL eines Unterknotens ersichtlich.

Jedem Informationsbedürfnis wird ein eindeutiger Code zugewiesen. Die Struktur der Codes ist wie in Listing 5.2 als *RDF-Triple* abgebildet. Über die *TERMS:hasTermChild*-Beziehung wird eine hierarchische Struktur erzeugt, die der Hierarchie der Informationsbedürfnisse entspricht.

Listing 5.2: Beispiel *RDF-Triple* der Informationsbedürfnisse in XML

```
1 <RDF xmlns:TERMS="http://mias.meduniwien.ac.at/TERMS/">
2 <Description about="http://mias.meduniwien.ac.at/TERM/0">
3   <TERMS:hastermChild resource="http://mias.meduniwien.ac.at/TERM/1"/>
4 </Description>
5 <Description about="http://mias.meduniwien.ac.at/TERM/1">
6   <TERMS:hastermChild resource="http://mias.meduniwien.ac.at/TERM/2"/>
7 </Description>
8 <Description about="http://mias.meduniwien.ac.at/TERM/1">
9   <TERMS:hastermChild resource="http://mias.meduniwien.ac.at/TERM/2.1"/>
10 </Description>
11 ...
12 </RDF>
```

Der Code 0 in Zeile 2 ist der Wurzelknoten. Dem Code 1 (i.e. Allergieanamnese) in Zeile 3 sind zwei Kinderknoten zugewiesen, in Zeile 6 Code 2 (i.e. Allergie) und in Zeile 9 Code 2.1 (i.e. Wirkstoff Kategorie).

Durch die eindeutigen URLs können die RDF-Dokumente in einem *Triple Store* vereint werden. Über die Beziehung *TERMS:hasTermBinding* und *TERMS:termBindingOf* werden die Terminologie-Codes aus Listing 5.2 mit den Archetyp-Knoten (siehe Listing 5.1) verbunden. Die textuelle Ausprägung wird dabei über die *ATS:labelDE* und *ATS:labelEN* den Terminologie-Codes zugewiesen. Im *Archetype Repository* wird nur der Text in der *ontology*-Sektion des Archetyps zur Namensgebung für ein Informationsbedürfnis herangezogen.

Zu den Datentypen werden nur einfache strukturelle Informationen wie zum Beispiel der konkrete XPath der jeweiligen Einheiten des Messwertes (i.e. *DATATYPE:dataTypeUnitPath*) oder der XPath zu den Werten selber (i.e. *DATATYPE:dataTypeValuePath*) hinterlegt. Diese werden für das Suchen in den EHR-Dokumenten benötigt und müssen abhängig vom zugrundeliegenden Referenzmodell und den verwendeten Datentypen angepasst werden.

### 5.1.2 Implementierung des *Archetype Repository*

Um das Erstellen des *Archetype Repository* zu vereinfachen, wurde dieses nicht manuell erzeugt, sondern mithilfe von Archetypen automatisch aufgebaut. Alle im Rahmen des EHR-Arche-Projekts erzeugten ISO/EN-13606-Archetypen (siehe Kapitel 4.1) wurden zum

Aufbau herangezogen.

Die Hierarchie der Informationsbedürfnisse wurde ebenfalls mit *RDF-Triple* dargestellt und mit den aus den Archetypen erzeugten *RDF-Triple* verbunden. Die *RDF-Triple* wurden dabei manuell aus den in Excel dokumentierten Informationsbedürfnissen erstellt. Die openEHR-Java-Referenz-Implementierung von [154] wurde als Ausgangspunkt verwendet, um Archetypen zu parsen. Als *Triple-Store* wurde das Java Jena Framework [155] verwendet. Es ermöglicht das Erzeugen von *RDF-Triple* sowie das Suchen und Navigieren darin. Die *RDF-Triple* wurden als XML-Dokumente serialisiert und werden beim Starten des *Archetype Repository* von dort geladen.

Das *Archetype Repository* wurde als Java-Webservice umgesetzt. Allen IHE-XDS-Akteuren wurden Schnittstellen zur Verfügung gestellt. Über unterschiedliche SOAP-Nachrichten werden die benötigten Informationen zwischen den Akteuren als XML ausgetauscht.

## 5.2 Umsetzung der inhaltsbasierten Suchanfragen

Die inhaltsbasierte Suchanfrage wird im erweiterten *Document Consumer* erzeugt und mittels Webservices an den *Document Crawler* gesendet.

Um Suchanfragen zu erzeugen, lädt der erweiterte *Document Consumer* beim Starten die hierarchische Liste aller Informationsbedürfnisse vom *Archetype Repository*. Diese Liste diente in weiterer Folge dazu, inhaltsbasierte Suchanfragen basierend auf der im nächsten Abschnitt beschriebenen XML-Struktur zu erzeugen.

Die Informationsbedürfnisse wurden, wie in Kapitel 5.1 beschrieben, mittels *RDF-Triple* im *Archetype Repository* abgebildet. Über den Webservice-Aufruf *getAllInfoItems* wird eine hierarchische Liste mit allen im *Archetype Repository* befindlichen Informationsbedürfnissen, die ident mit den Suchbegriffen sind, erzeugt. Eine Liste aller Informationsbedürfnisse kann unter [http://www.meduniwien.ac.at/msi/arche/sites/default/files/Info%20Items%20englisch\\_0.pdf](http://www.meduniwien.ac.at/msi/arche/sites/default/files/Info%20Items%20englisch_0.pdf) heruntergeladen werden. Jedem Suchbegriff wurde ein eindeutiger Code zugewiesen, die Hierarchie der Codes entspricht der Struktur der Informationsbedürfnisse und wird beim Erzeugen der Suchanfrage verwendet. In Listing 5.3 wird ein Ausschnitt der Rückgabe des *Archetype Repository* für das Informationsbedürfnis “Körpergewicht” gezeigt. Das “Körpergewicht” befindet sich hierarchisch unterhalb des

Listing 5.3: Auszug der Suchbegriffe vom *Archetype Repository*

```
1 <?xml version="1.0"?>
2 <node localCode="56" labelDe="Selbstmessungen des Patienten" labelEn="Self
  -monitoring of the patient">
3   <element localCode="3.1" labelDe="Koerpergewicht" labelEn="body weight">
4     <PQ>
5       <unit>kg</unit>
6     </PQ>
7   </element>
8   <node localCode="4" labelDe="Blutglukose" labelEn="blood glucose">
9     <node localCode="116" labelDe="Postprandialer 2-Stunden-Blutglucosewert"
      labelEn="Two- hour postprandial plasma glucose"/>
10    <node localCode="117" labelDe="Blutglukose nuechtern" labelEn="blood
      glucose fasting"/>
11  </node>
12 </node>
```

Knotens “Selbstmessung des Patienten”, ist vom Datentyp *Physical Quantity* (PQ) und wird in der Einheit Kilogramm (kg) angegeben.

In der Suchanfrage wurde der zurückgelieferte Wert des Attributes *localCode* als eindeutiger Code für den Suchbegriff verwendet. Die Attribute *labelDe* und *labelEn* beinhalten die deutsche und englische Beschriftung zum jeweiligen Code. Dem Benutzer wird damit die Beschriftung in der jeweils gewünschten Sprache angezeigt. Stellt ein Informationsbedürfnis eine Ausprägung eines elementaren Wertes dar, wird es als XML-Element `<element>` dargestellt und nicht wie sonst als XML-Element `<node>`. Die XML-Elemente `<element>` beinhalten Angaben zum zugrundeliegenden Datentyp und den erlaubten Ausprägungen. Diese Informationen wurden innerhalb der Suchanfrage im XML-Element `<Value>` oder `<Unit>` verwendet.

Basierend auf dieser Liste kann im *Document Consumer* eine thesaurusbasierte Suchanfrage erzeugt werden (siehe Abbildung 5.4). Es können zwei Arten von Suchanfragen unterschieden werden. Die Ad-hoc-Suchanfragen bestehen aus einem einzelnen Suchbegriff, die vorgefertigte Suchanfrage kann aus beliebig vielen dieser Suchbegriffe zusammengesetzt werden. Zu jeder Suchanfrage kann eine Zeiteinschränkung hinzugefügt werden.

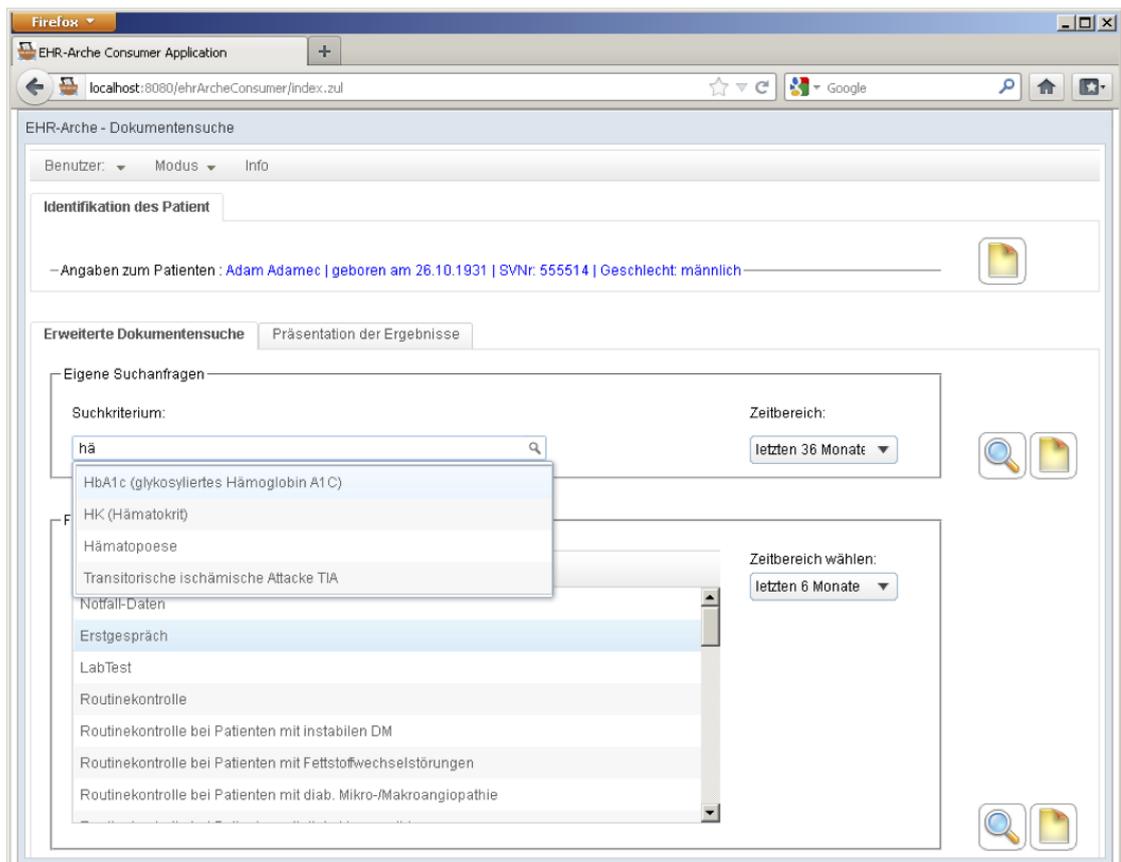


Abbildung 5.4: Benutzeroberfläche zum Erzeugen der inhaltsbasierten Suchanfragen im erweiterten *Document Consumer*.

### 5.2.1 XML-Struktur der inhaltsbasierten Suchanfrage

Für das Abbilden der inhaltsbasierten Suchanfrage wurde eine eigene XML-Struktur entwickelt. Die hier vorgestellte XML-Struktur erlaubt es, Begriffe des Thesaurus beliebig zu verschachteln und diese mit booleschen Operatoren zu verknüpfen. Anhand der in Listing 5.4 gezeigten Suchanfrage wird der Aufbau der XML-Struktur vorgestellt.

Eine inhaltsbasierte Suchanfrage ist immer nach dem folgenden Schema aufgebaut. Da es sich um eine XML-Struktur handelt, muss genau eine Wurzelklasse vorhanden sein, die die weiteren XML-Elemente umschließt. Als Wurzelklasse dient `<SearchQuery>`, wie in Zeile 1 in Listing 5.4 ersichtlich. In der Suchanfrage werden alle Dokumente, die nach dem 4. April 2011 erzeugt wurden, gesucht, die Werte für systolischen Blutdruck größer  $139\text{ mm[HG]}$  und

diastolischen Blutdruck größer 80 *mm[HG]* oder einen *HbA1c*-Wert beinhalten. Die Werte sind wie im Folgenden beschrieben mit UND und ODER verknüpft:

$$(\textit{Zeitbereich} > 4. \textit{April} 2011) \wedge ((\textit{systolisch} > 139 \textit{ mm[HG]}) \wedge (\textit{diastolisch} > 80 \textit{ mm[HG]})) \vee \textit{HbA1c} .$$

Der `<MetaData>`-Abschnitt der Suchanfrage (Zeile 2–5) beinhaltet die für die Umsetzung der metadatenbasierten Suche relevanten Metadaten wie die Identifikation des Arztes und die Patienten-ID. Weitere Metadaten wie Zeitstempel, die etwa für das Loggen der Suchanfrage benötigt werden etc., können bei Bedarf in diesem Teil angegeben werden. Die in der metadatenbasierten Suchanfrage angegebenen Zeitbereiche werden in der hier vorgestellten Methode nicht als Metadaten gesehen, sondern zu den jeweiligen Begriffen hinzugefügt, wie im Folgenden beschrieben.

Das XML-Element `<TUPLE>` (z.B. Zeile 16–21) beinhaltet die Referenz auf den zu suchenden Begriff. In der XML-Struktur werden die Suchbegriffe nicht direkt als Text angegeben, sondern über den in der Wissensbasis hinterlegten Code im darunterliegenden XML-Element `<ItemCode>` (z.B. Zeile 17). Werden in der Wissensbasis unterschiedliche medizinische Terminologien verwendet, müsste im `<ItemCode>` zusätzlich zum verwendeten Code noch das verwendete Codesystem bzw. die verwendete Terminologie angegeben werden.

Neben der Verknüpfung der einzelnen Suchbegriffe können abhängig vom Datentyp, der über die Wissensbasis einem Suchbegriff zugeordnet ist, zusätzlich Einschränkungen gemacht werden. Bei Suchanfragen, die nur einen einzelnen Suchbegriff enthalten wie zum Beispiel “systolischer Blutdruck” oder “HbA1c-Wert”, kann der gewünschte Wertebereich eingeschränkt bzw. angegeben werden. Ist der Suchbegriff mit einem weniger spezifischen Informationsbedürfnis wie zum Beispiel Blutdruckmessung verknüpft, sind diese zusätzlichen Einschränkungen nicht möglich. Diese weiteren Einschränkungen werden über die XML-Elemente `<Value>`, `<Operator>` und `<Unit>` (z.B. Zeile 18–20) angegeben. `<Value>` gibt die Ausprägung des EHR-Inhalts, der durch den Begriff beschrieben wird, an. Die `<Unit>` (Zeile 24) gibt die physikalische Einheit an, nach der die Suche eingeschränkt werden soll. Der `<Operator>` (Zeile 25) beinhaltet den Operator, um die Werte einzuschränken. Mögliche Operatoren sind

- gleich (EQ engl. *equal to*)
- nicht gleich (NE engl. *not equal to*)
- größer (GT engl. *greater than*)
- größer gleich (GE engl. *greater than or equal to*)
- kleiner (LT engl. *less than*)
- kleiner gleich (LE engl. *less than or equal to*)

Eine <TUPLECOLLECTION> (Zeile 6) erlaubt es, mehrere <TUPLE> miteinander zu verknüpfen. Eine <TUPLECOLLECTION> hat immer ein Attribut *operator* mit dem Wert *AND* bzw. *OR*. Der *operator* dient dazu, weitere <TUPLECOLLECTION> oder <TUPLE> mit UND und ODER zu verbinden. Die <TUPLECOLLECTION> enthält immer mindestens zwei Kinder-Elemente (i.e. <TUPLECOLLECTION>, <TUPLE>).

Die hier präsentierte XML-Struktur erlaubt es, neben der Verknüpfung der Suchbegriffe diese zusätzlich abhängig vom Typ einzuschränken. Die Einschränkungen setzen jedoch voraus, dass zusätzliche Informationen über die Semantik der einzelnen Begriffe vom Thesaurus mitgeschickt werden bzw. im Thesaurus auch vorhanden sind. Die Datentypen der EHR-Inhalte, die mit einem Begriff referenziert sind, wurden für die Auswahl der erlaubten Operatoren verwendet. Für Text-Ausprägungen standen die beiden Operatoren *gleich* und *ungleich* zur Verfügung. Für numerische Werte und Zeitangaben konnten zusätzlich die Operatoren *größer*, *größer gleich*, *kleiner*, *kleiner gleich* verwendet werden. Zeitbereiche wurden über zwei <TUPLE> in einer <TUPLECOLLECTION> mit dem Attribut UND abgedeckt, wobei bei einem *größer* und beim anderen *kleiner* angegeben wurde. Das Zeitformat musste dem XML-Datentyp *DateTime* (i.e. YYYY-MM-DD'T'hh:mm:ss) entsprechen. Für offene Zeitbereiche ab einem gewissen Datum bzw. bis zu einem gewissen Datum wurde nur eine Zeiteinschränkung angegeben (siehe Zeile 8–11). Waren die möglichen Einheiten der eingeschränkten Begriffe bekannt und wurden diese in der <Unit> angegeben, wurde nur explizit nach Werten mit dieser Einheit gesucht. Eine Umrechnung zwischen den Einheiten (z.B. Zentimeter auf Meter) könnte durch diese Modellierung bei der Suche zusätzlich umgesetzt werden.

Listing 5.4: Suchanfrage für systolischen UND diastolischen Blutdruck ODER Hba1c

```
1 <SearchQuery>
2   <MetaData>
3     <author>Dr. TestArzt</author>
4     <patid>555514</patid>
5   </MetaData>
6   <TUPLECOLLECTION operator="AND">
7     <TUPLECOLLECTION operator="AND">
8       <TUPLE>
9         <ItemCode>999</ItemCode>
10        <Operator>GT</Operator>
11        <Value>2011-04-01T09:54:23</Value>
12      </TUPLE>
13    </TUPLECOLLECTION>
14    <TUPLECOLLECTION operator="OR">
15      <TUPLECOLLECTION operator="AND">
16        <TUPLE>
17          <ItemCode>5.1</ItemCode>
18          <Unit>mm[HG]</Unit>
19          <Operator>GT</Operator>
20          <Value>139</Value>
21        </TUPLE>
22        <TUPLE>
23          <ItemCode>5.2</ItemCode>
24          <Unit>mm[HG]</Unit>
25          <Operator>GT</Operator>
26          <Value>80</Value>
27        </TUPLE>
28      </TUPLECOLLECTION>
29    </TUPLECOLLECTION operator="AND">
30      <TUPLE>
31        <ItemCode>86.6</ItemCode>
32      </TUPLE>
33    </TUPLECOLLECTION>
34  </TUPLECOLLECTION>
35 </SearchQuery>
```

Um die Suchanfrage so einfach wie möglich zu halten, wurde nicht zwischen Werten, die eingeschränkt werden, und solchen, die im Resultat dann auch angezeigt werden, unterschieden. In SQL-Suchanfragen der Form `SELECT a FROM x WHERE a=1 AND b=2 OR c=3` könnte diese Unterscheidung gemacht werden. In der SQL-Suchanfrage werden nur die Zeilen ausgewählt, in denen die Werteinschränkungen der WHERE-Klausel für die Spalten a, b und c erfüllt sind. Mittels des SELECT-Befehls könnte zusätzlich angegeben werden, welche Spalten im Resultat angezeigt werden sollen, in diesem Fall nur Spalte a.

### 5.2.2 Das Suchinterface

In der inhaltsbasierten Suchanfrage vom GDA konnten keine strukturellen Einschränkungen für die gesuchten Daten gemacht werden. Die strukturellen Informationen wurden stattdessen beim Auflösen der Suchanfrage im *Document Crawler* automatisch hinzugefügt. Dies hat den weiteren Vorteil, dass referenzmodellunabhängige Suchanfragen vom GDA formuliert werden können. Die strukturellen Informationen zum jeweiligen Referenzmodell wurden ebenfalls erst beim Auflösen der Suchanfrage berücksichtigt.

In Abbildung 5.4 ist die Oberfläche zum Absenden der inhaltsbasierten Suchanfragen im erweiterten *Document Consumer* zu sehen. Die 404 eindeutigen Suchbegriffe können über ein Suchfeld mit Autovervollständigung gesucht werden. In [156] werden unterschiedliche Suchoberflächen vorgestellt. Speziell bei Suchen, die länger dauern, wird empfohlen, Autovervollständigung zu verwenden. Dies verringert die Anzahl der möglichen Suchbegriffe und gibt dem Benutzer eine Vorstellung, welche Suchanfragen zu Ergebnissen führen. Bei der Ad-hoc-Suche kann jeweils nur ein einziger Suchbegriff ausgewählt werden. Nachdem ein Zeitbereich für die Suche festgelegt wird, kann die Ad-hoc-Suchanfrage an den *Document Crawler* gesendet werden. Zusätzlich wurden 30 vorgefertigte Suchanfragen für komplexere Suchanfragen entwickelt, die mehrere Suchbegriffe beinhalten. Ein Beispiel einer solchen Suchanfrage ist die Suchanfrage "Notfall-Daten", die aus den Suchbegriffen "Allergiediagnose" (Code 1) ODER "Therapie-Nebenwirkungen" (Code 103) besteht, oder die Suchanfrage "Erstgespräch", die aus 86 ODER-verknüpften unterschiedlichen Suchbegriffen besteht. Die vorgefertigten Suchanfragen erlauben es dem GDA, schnell einen Überblick über einen neuen Diabetes-Patienten zu bekommen.

Wie in [157, 158] beschrieben, wurde im Rahmen des EHR-Arche-Projekts ein Prototyp zur

Erzeugung der erweiterten Suchanfragen (siehe Abbildung 5.5) erstellt. Der Suchanfragen-Editor erlaubt es, beliebig viele Suchbegriffe mit UND und ODER und Klammern zu verknüpfen. Diese Suchanfragen können als vorgefertigte Suchanfragen gespeichert werden oder direkt an den *Document Crawler* gesendet werden. Der Suchanfragen-Editor wurde im Rahmen des Projektes nicht evaluiert und wurde von den Ärzten nicht verwendet. Es ist aber ersichtlich, dass die beschriebene Suchanfrage mit einer graphischen Benutzeroberfläche erzeugt werden kann. Über die visuelle Oberfläche ist es möglich, die Hierarchie der Suchbegriffe darzustellen. Dies erlaubt es, den Kontext eines einzelnen Codes besser erkennen zu können und dieses Wissen beim Erzeugen der Suchanfrage einfließen zu lassen.

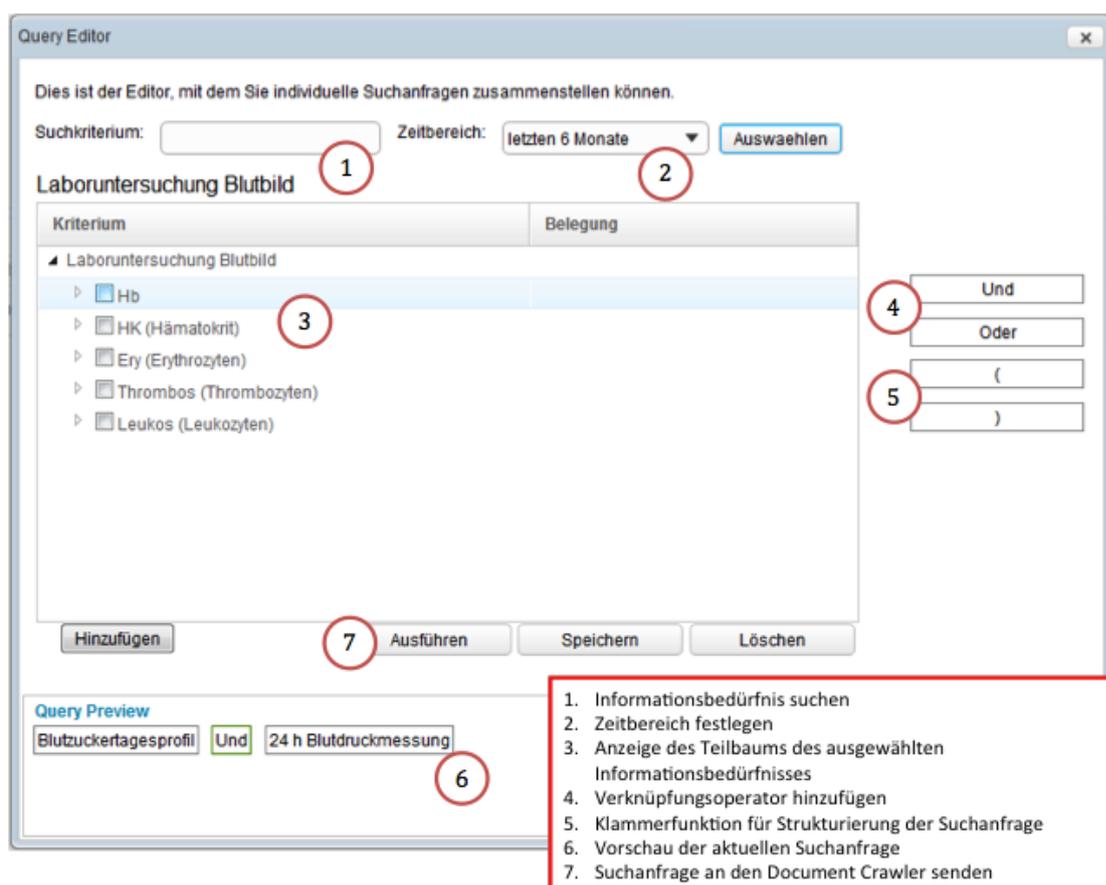


Abbildung 5.5: Suchanfragen-Editor im erweiterten *Document Consumer* (aus [158]).

### 5.3 Erzeugen der IHE-XDS-Metadatensuche

Um aus der Suchanfrage des *Document Consumers* eine metadatenbasierte Suchanfrage zu erzeugen, musste die Suchanfrage im ersten Schritt geparkt werden. Mithilfe des *Archetype Repository* wurde im zweiten Schritt die metadatenbasierte Suchanfrage erstellt.

Die Suchanfrage des *Document Consumers* wurde über eine Webservice-Nachricht an den *Document Crawler* gesendet. Die Suchanfrage wurde im *Document Crawler* geparkt und als Java-Objekt instanziiert. Zum Parsen der XML-Dokumente wurde die Open-Source-JAVA-DOM-Implementierung JDOM verwendet. Beim Parsen wurde jedes XML-Element der Suchanfrage in eine Instanz der generischen Klasse *QueryNode* umgewandelt. Das Parsen war abgeschlossen, wenn die Suchanfrage des *Document Consumers* vollständig als Java-Instanz vorlag. Die Klasse *QueryNode* hat folgende Eigenschaften:

- Vier Typen von *QueryNode* werden unterschieden (i.e. "METADATA", "SEARCHQUERY", "TUPLECOLLECTION" und "TUPLE").
- Jede *QueryNode* hat einen Verweis zur Eltern-*QueryNode*.
- Jede *QueryNode* verweist auf alle Kinder-*QueryNodes*.
- Zu jeder *QueryNode* können ein Wert, der Operator und Einheit sowie der Code des zugrundeliegenden Informationsbedürfnisses gespeichert werden.

Für das Erzeugen der metadatenbasierten Suche wurden neben den in der Suchanfrage explizit vorkommenden Informationen wie Patienten-ID, Zeitbereiche und GDA-ID die relevanten Dokumenttypen benötigt. Da nicht alle Codes in allen Dokumenttypen vorkommen, konnten gewisse Dokumenttypen von vornherein von der Suche ausgeschlossen werden, um die Suche zu beschleunigen. Im Standard ISO/EN 13606 können Dokumenttypen mit COMPOSITION-Archetypen gleichgesetzt werden. Die Dokumenttypen konnten somit aus den verwendeten Codes in den *QueryNodes* berechnet werden. Zu jedem Code lieferte das *Archetype Repository* die verknüpften Archetyp-Knoten und die COMPOSITION-Archetypen, die diese Archetyp-Knoten beinhalten, zurück. Die gefundenen COMPOSITION-Archetypen entsprachen den relevanten Dokumenttypen.

In Abbildung 5.6 ist ein Ausschnitt des *Archetype Repository* zu sehen. Er zeigt den Archypen “Diabetesklassifikation” mit den damit verbundenen Archypen. Die “Diabetesklassifikation” wird in den unterschiedlichen “Arztbriefen” (Discharge\_summarization\_note...), im “Konsiliarbefund” und in der “Medizinischen Aufnahme/Anamnese” verwendet. Die “Diabetesklassifikation der Familienanamnese” ist eine Spezialisierung der “Diabetesklassifikation”. Wird der Suchbegriff “Diabetesklassifikation” angegeben, werden die hier angezeigten Dokumententypen zurückgeliefert.

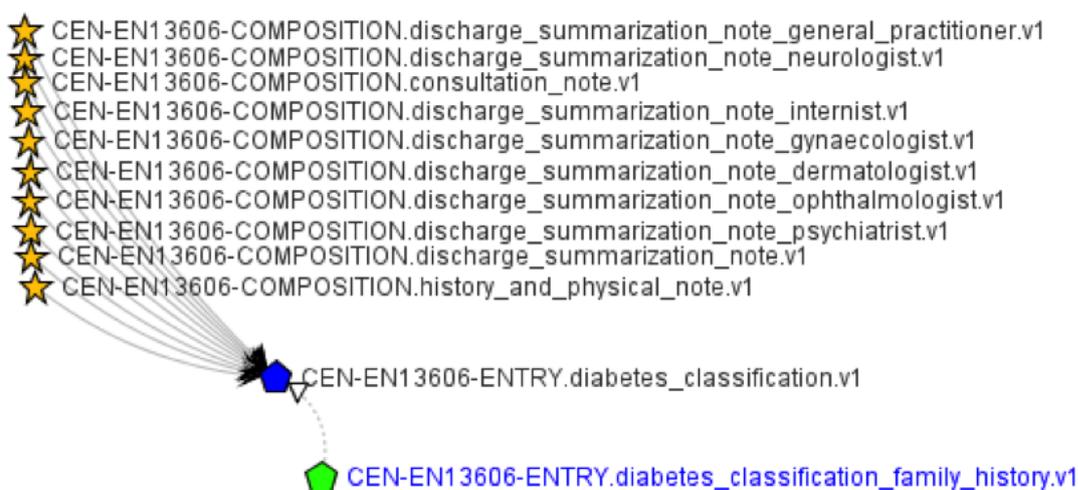


Abbildung 5.6: “Diabetesklassifikation” im *Archetype Repository* mit den damit verbundenen Archypen.

Bei einer herkömmlichen metadatenbasierten Suche wird nur ein einziger Zeitbereich als Suchparameter angegeben. Da bei der inhaltsbasierten Suche zu jedem Suchbegriff ein Zeitbereich angegeben werden kann, kann eine optimierte metadatenbasierte Suche angeboten werden. Für jeden in der inhaltsbasierten Suche definierten Zeitbereich wurde eine metadatenbasierte Suche generiert. Alle so erhaltenen metadatenbasierte Suchen wurden an die *Document Registry* gesendet. Die Antworten der *Document Registry* wurden zu einer einzigen Antwort aggregiert. Durch diese Aggregation wurden Dokumente, die durch mehrere metadatenbasierte Suchen gefunden werden, dem GDA nur einmal angezeigt.

Bei der metadatenbasierten Suche im verwendeten IHE-XDS-Framework war es nicht möglich, die Fachrichtung des Arztes, der die Dokumente hochgeladen hat, direkt von der *Document*

*Registry* durchsuchen zu lassen. Dies wurde zum Beispiel benötigt, um einen Arztbrief vom Allgemeinmediziner von einem vom Internisten zu unterscheiden. Um eine Suche über die unterschiedlichen Arztbriefe zu ermöglichen, wurden die Metainformationen zu den jeweiligen EHR-Dokumenten von der *Document Registry* heruntergeladen und der darin gespeicherte “practice\_setting\_code” im *Document Crawler* ausgewertet. Diese Information floss in die im vorherigen Schritt erzeugte aggregierte Antwort zusätzlich ein.

## 5.4 Erzeugen der XQuery-Suchanfrage

Um die metadatenbasierte Suche zu erzeugen, wurde die Suchanfrage des *Document Consumers* als *QueryNodes* abgebildet. Diese *QueryNodes* wurden ebenfalls verwendet, um eine XQuery der Suchanfrage zu erzeugen.

Die unterschiedlichen Schritte beim Erzeugen der XQuery-Suchanfrage werden im Folgenden im Detail beschrieben und mittels des Beispiels in Listing 5.4 auf Seite 113 erläutert.

### 5.4.1 Auflösen der Suchanfrage

Die im *Document Consumer* erzeugte Suchanfrage besteht aus einer hierarchischen Liste von Suchbegriffen, die mit den booleschen Operatoren UND und ODER verknüpft sind und zusätzliche Werteinschränkungen beinhalten können. Beim Auflösen der Suchanfrage wurden die Suchbegriffe mithilfe der Information aus dem *Archetype Repository* in Relation zu XML-Elementen in EHR-Extrakten gesetzt. Über die strukturelle Information wurde die Granularität der Rückgabewerte ermittelt sowie die XQuery erzeugt. Zum Auflösen der Suchanfrage wurden die Suchbegriffe der inhaltsbasierten Suchanfrage gegen Referenzen zu den konkreten Knoten ausgetauscht. Kam ein Suchbegriff in mehreren Dokumenttypen oder innerhalb eines Dokumenttyps öfter vor, wurde für jeden Knoten eine separate Referenz erstellt.

Für jeden Suchbegriff wurden mithilfe des Archetyp-Repository die dazugehörigen Archetyp-Knoten gesucht. Durch die Verwendung des Archetyp-Repository bei der Suchanfragenerzeugung konnte jeder Suchbegriff mindestens einem Archetyp-Knoten zugewiesen werden. Durch das Wiederverwenden von Archetypen mithilfe des *Slot*-Mechanismus kamen die meisten Suchbegriffe in genau einem Archetyp vor. Ein medizinisches Konzept sollte jeweils nur von einem einzelnen Archetyp beschrieben werden.

Eine Ausnahme bilden die Spezialisierungen von Archetypen wie zum Beispiel “Laborbefund” und “Laborbefund Glukosestatus”. Die “Analyse” (siehe Listing 2.1 Code 86.8 auf Seite 19) wird in beiden Fällen mit demselben Suchbegriff beschrieben, nur der Kontext ändert sich. Die beiden folgenden eindeutigen URLs referenzieren die “Analyse” in unterschiedlichen Archetypen:

- [http://mias.meduniwien.ac.at/ATS/CEN-EN13606-ENTRY.lab\\_test\\_Glucose\\_status.v1/items%3Cat0008.1%3E](http://mias.meduniwien.ac.at/ATS/CEN-EN13606-ENTRY.lab_test_Glucose_status.v1/items%3Cat0008.1%3E)
- [http://mias.meduniwien.ac.at/ATS/CEN-EN13606-ENTRY.lab\\_test.v1/parts%3Cat0008%3E](http://mias.meduniwien.ac.at/ATS/CEN-EN13606-ENTRY.lab_test.v1/parts%3Cat0008%3E)

Zu jedem Suchbegriff wurde eine Referenz zum Archetyp-Knoten in der Java-Instanz (i.e. *QueryNode*) gespeichert. Kam ein Suchbegriff in mehreren Archetypen vor, wurde der Klassenbaum um eine `<TUPLECOLLECTION>` mit dem Operator ODER erweitert, die für jeden Archetyp-Knoten eine Kopie der ursprünglichen Klasse des Suchbegriffs mit Referenz auf den Archetyp-Knoten enthielt. Alternativ hätte jedem Suchbegriff eine Liste mit den Archetyp-Knoten angehängt werden können. Da diese Spezialfälle jedoch auch schon beim Erzeugen der Suchanfrage auftreten können, wurde ein Weg gesucht, der die ursprüngliche Methodik der Suchanfrage nicht bricht.

Einen weiteren Spezialfall stellten Suchbegriffe dar, die nicht einen Archetyp-Knoten, sondern eine Ausprägung dieses Knotens beschreiben. Beim Beispiellarchetyp in Listing 2.1 auf Seite 19 ist der Knoten *at0064* vom Datentyp `SIMPLE_TEXT`. Die zugelassenen Ausprägungen sind mit den beiden Codes *at0840* und *at0844* angegeben, der Suchbegriff “HbA1c” mit dem Code *AAArche::86.6* ist mit dem Knoten *at0844* verknüpft. Verweist der Suchbegriff auf eine textuelle Ausprägung, wurde der Archetyp-Knoten (in diesem Fall *at0064*) als neuer Knoten für den Suchbegriff verwendet, wobei eine Referenz zum ursprünglichen Knoten zusätzlich noch bei der *QueryNode* gespeichert wurde. Die textuelle Ausprägung von Archetyp-Knoten (i.e. *at0844*) wurde als Value in der *QueryNode* mit dem Operator EQ gespeichert. Waren für einen Knoten mehrere Übersetzungen in der *ontology*-Sektion des Archetyps definiert, wurde für jede Sprache eine eigene *QueryNode* erzeugt und mit ODER mit den anderen Knoten verknüpft.

## 5.4.2 Auflösen der UND- und ODER-Verknüpfungen

Die booleschen Operatoren der Suchanfrage wurden wie in Kapitel 3.1.4 beschrieben aufgelöst. Zusätzlich zu den Containern, die bei den UND- und ODER-Suchanfragen zurückgeliefert wurden, wurde die UND- und ODER-Hierarchie in eine disjunktive Normalform (DNF) transformiert. Eine DNF ist eine Disjunktion von Konjunktionen. Die in DNF konvertierte Suchanfrage konnte direkt in eine XQuery-Suchanfrage transformiert werden, wie im nächsten Abschnitt beschrieben wird. Die Suchanfrage aus Listing 5.4 kann als vereinfachte Formel so dargestellt werden:  $999 \wedge ((5.1 \wedge 5.2) \vee 86.6)$ . Nach der Konvertierung in DNF erhalten wir  $(999 \wedge 5.1 \wedge 5.2) \vee (999 \wedge 86.6)$ .

## 5.4.3 Festlegen der Minimal Information Unit

Bei der Suche ist es wichtig zu wissen, welche Art von Archetyp-Knoten mit dem jeweiligen Suchbegriff verknüpft ist, um festzustellen, welche Daten dem GDA als Ergebnis angezeigt werden sollen.

Die Klassen COMPOSITION, SECTION, ENTRY sind Hüllenklassen, die zur eindeutigen Interpretation nicht von anderen Klassen abhängig sind. Bezieht sich ein Suchbegriff auf diese Klassen, kann direkt ein aussagekräftiger Rückgabewert zurückgeliefert werden. Es besteht jedoch die Gefahr, dass die zurückgelieferten Fragmente zu unspezifisch sind. Dem Benutzer können in diesem Fall schon beim Erzeugen der Suchanfrage Suchbegriffe, die hierarchisch unter seinem Suchbegriff liegen, angeboten werden, um die Suche spezifischer zu machen. Ansonsten muss davon ausgegangen werden, dass der gewählte Suchbegriff der gewünschten Granularität entspricht.

Die Klasse CLUSTER kann verwendet werden, um Tabellen und Tabellen-Spalten in EHR-Dokumenten abzubilden. Bezieht sich ein Suchbegriff auf einen einzelnen CLUSTER oder ein ELEMENT eines CLUSTERS mit dem *structure\_type*-Attribute "STRC02", wurde die ganze Tabelle als Rückgabe zurückgeliefert.

Bei der ELEMENT-Klasse kann es sich wie bei der Klasse CLUSTER schon erwähnt um einen Wert in einer Zeile einer Tabelle handeln. Wird nur das Element zurückgeliefert, kann der Kontext verloren gehen, zum Beispiel wird der Kopf der Spalte nicht zurückgeliefert. In Abbildung 5.7 ist ein Beispiel einer Ausprägung zu sehen ("LDL-Cholesterin"), die auch als Suchbegriff verwendet werden kann. Beim Auflösen der Suchanfrage wurden textuelle

Ausprägungen automatisch auf deren Elternknoten referenziert. Dies würde im Beispiel von Abbildung 5.7 für die *Minimal Information Unit* bedeuten, alle Analyse-Elemente zurückzuliefern, was in einer Liste mit “Analyse = LDL-Cholesterin” resultieren würde. Gewünscht sind jedoch die Ausprägungen der Analyse, die in diesem Fall unter “Ergebnis” gespeichert sind. Um alle “LDL-Cholesterin”-Werte zu suchen, durfte in weiterer Folge nicht nur der Wert Analyse, sondern es musste zusätzlich der darüberliegende Knoten (in diesem Fall “CLUSTER”) gesucht werden.

Laboruntersuchung Lipidstatus	
CLUSTER	
Analyse	LDL-Cholesterin
Ergebnis value	165
unit	mg/dl

Abbildung 5.7: Formularelemente für Dokumentation von LDL-Cholesterin.

In [159] werden konkrete Ansätze zum Eruiieren der Reihenfolge der Suchergebnisse in strukturierten XML-Dokumenten angesprochen. Um den Kontext zu erhalten, wird dabei auch auf Elternknoten zurückgegriffen. Da dieser Sachverhalt nicht nur bei CLUSTERS auftritt, die als Tabelle identifiziert sind, wurde folgender Algorithmus gewählt, um die *Minimal Information Unit* zu finden:

War ein Suchbegriff einer textuellen Ausprägung im Archetyp zugewiesen und hatte der Elternknoten dieses Archetyp-Knotens nur ELEMENTs als Geschwisterknoten, wurde der Elternknoten als *Minimal Information Unit* angenommen und als Suchkontext verwendet.

#### 5.4.4 Auflösen der Werteinschränkungen

Im nächsten Schritt wurden die Werteinschränkungen in den Suchanfragen aufgelöst. Zu jedem Suchbegriff wurden vom *Archetype Repository* mögliche Werteinschränkungen abhängig vom zugrundeliegenden Datentyp mitgeliefert. In der Suchanfrage wurden diese Werte dann in den Elementen <Value> und <Operator> im <TUPLE> verwendet. Abhängig vom zugrundeliegenden Datentyp und den Operatoren wurden die Werteinschränkungen in unterschiedliche XPath-Ausdrücke umgewandelt.

Beim Empfangen der Suchanfrage wurde durch eine Anfrage an das *Archetype Repository* zu jedem Suchbegriff der zugrundeliegende Datentyp eruiert. Nur Suchbegriffe, die einem ELEMENT-Archetyp-Knoten zugewiesen waren, wurden im Folgenden berücksichtigt. Es wäre auch möglich, Werteschränkungen z.B. auf ENTRY-Ebene zuzulassen, um das *subject\_of\_information* (siehe Kapitel 2.1.1) einzuschränken; diese Spezialfälle wurden jedoch in der vorliegenden Arbeit nicht berücksichtigt.

Betraff die Werteschränkung einen Suchbegriff, der ein ELEMENT vom Datentyp SIMPLE\_TEXT beschreibt, wurde die Werteschränkung in einen XPath-Ausdruck umgewandelt. War die Werteschränkung durch einen Suchbegriff, der eine Ausprägung eines Archetyp-Knotens beschrieb, zustande gekommen, wurden zwei separate Knoten mit der jeweiligen Übersetzung erzeugt. Dieser Spezialfall musste beim Auflösen der Werteschränkung nicht mehr speziell berücksichtigt werden. Bei dieser Werteschränkung waren nur die Operatoren *gleich* oder *ungleich* möglich und sie wurden nach folgendem Muster in einen XPath-Ausdruck umgewandelt:

- Pfad vom ELEMENT bis zum Ort, an dem die Werteschränkung angewendet werden muss
- die Operatoren gleich/ungleich
- die Werteschränkung in einfachen Hochkommas

Beispiele für XPath-Ausdrücke für textuelle Einschränkungen:

- `/*:value/*:originalText = 'Triglyceride'`
- `/*:value/*:originalText [contains(.,'Rauch')]`

Die Elemente zwischen den Schrägstrichen beziehen sich auf Knoten im XML-Dokument, die in diesen Beispielen Klassen und Attributen des Referenzmodells bzw. den zugrundeliegenden Datentypen entsprechen. Der Ausdruck “\*:” besagt, dass alle *Namespaces* im XPath-Ausdruck berücksichtigt werden sollen. Die Werteschränkungen beim Datentyp SIMPLE\_TEXT beinhalteten nur die vorgegebenen Suchbegriffe. Der Suchbegriff entsprach immer genau dem Begriff im EHR-Dokument und ein XPath-Ausdruck mit = bzw. != war ausreichend. Um

Werteinschränkungen, die direkt vom Benutzer eingegeben werden, umzusetzen, könnte die XPath-Funktion *contains(string1, string2)* verwendet werden. *Contains* überprüft, ob *string2* in *string1* vorkommt.

Werteinschränkungen auf Datums-Felder wurden nach demselben Muster umgesetzt. Im Gegensatz zu den textuellen Einschränkungen waren hier alle Operatoren erlaubt. Um Intervalle abzubilden, konnte entweder nur einer der Werte angegeben werden oder zwei Zeiteinschränkungen mit UND verknüpft mit jeweils einem Größer- und einem Kleiner-Operator. Ein Beispiel für die XPath-Ausdrücke zum Einschränken eines Zeitpunkts ist im Folgenden zu sehen.

- `/*:time > '2012-03-11T10:07:00'`

Für numerische Werte dürfen keine Anführungszeichen verwendet werden, da diese sonst als *String* interpretiert werden. Auch hier können Intervall-Einschränkungen durch UND-Verknüpfungen der unteren und der oberen Grenze des Intervalls gemacht werden. Ein Beispiel eines numerischen Ausdrucks kann wie folgt aussehen.

- `/*:value/*:value > 200`

Die abhängig von den zugrundeliegenden Datentypen erhaltenen XPath-Ausdrücke wurden direkt in der jeweiligen *QueryNode* gespeichert. Der XPath-Ausdruck wurde dabei nicht in ein eigenes Feld geschrieben, sondern ersetzte direkt den ursprünglichen Wert im `<value>`-Feld. Das Hinzufügen eines separaten Feldes wäre jedoch sinnvoll, um den originalen Wert der Suchanfrage nicht zu verlieren. Dies wurde jedoch nicht gemacht, da die XML-Struktur im *Document Consumer* schon umgesetzt war und dort zusätzliche Änderungen mit sich gebracht hätte.

#### **5.4.5 Generierung der XQuery-Suchanfrage**

Im nächsten Schritt wurde die geparte Suchanfrage in eine XQuery-Suchanfrage umgewandelt. Es wurde pro inhaltsbasierter Suchanfrage eine XQuery-Suchanfrage erzeugt, die auf die EHR-Dokumente angewendet werden kann. Das vorangegangene schrittweise Auflösen der Suchanfragen war speziell auf das Erzeugen einer XQuery-Suchanfrage

ausgerichtet. Speziell das Auflösen in eine DNF ermöglicht es, aus jeder Konjunktion einen XQuery-FLOWR-Ausdruck zu erzeugen.

Das Ergebnis einer XQuery-Suchanfrage, die auf ein EHR-Dokument angewendet wird, wurde mittels eines (im nächsten Abschnitt beschriebenen) einzelnen `<row>`-Element repräsentiert. Für jede Konjunktion wurde eine Variable nach dem Schema `$queryX` angelegt, wobei das `X` die eindeutige Referenz für eine Konjunktion ist (i.e. "columnNumber"). Für jede Konjunktion wurde so eine eigene "columnNumber" erzeugt. Die Variable wurde mit einem FLOWR-Ausdruck befüllt, der die einzelnen Begriffe der jeweiligen Konjunktion berücksichtigt. Die unterschiedlichen Variablen wurden mit der XQuery-Funktion `functx:distinct-deep` [160] in eine einzelne `<row>` zusammengefügt. Die Funktion `functx:distinct-deep` überprüft, ob ein Knoten doppelt existiert (i.e. alle Unterknoten haben genau dieselben Attribute und dieselben Unterknoten mit denselben Werten), und liefert diese nur einmal zurück. Die Funktion `functx:is-node-in-sequence-deep-equal`, die ebenfalls in der XQuery definiert ist, wird von der Funktion `functx:distinct-deep` zur Überprüfung benötigt.

Die FOR-Klausel bekam den gemeinsamen Elternknoten der Konjunktionen als Ausgangspunkt. Für alle gefundenen Elternknoten wurden die in der WHERE-Klausel angegebenen Einschränkungen überprüft. Waren die Einschränkungen erfüllt, wurde der Elternknoten zurückgegeben. Im RETURN-Statement wurden noch die Spaltennummern mit der Funktion `functx:add-attributes` hinzugefügt. Es war hier auch möglich einen Flag zu setzen, falls in einem über dem Elternknoten liegenden Element (*Ancestor*) Information angegeben werden soll. Im dargestellten Beispiel im nächsten Abschnitt wurde das Attribut `otherSubject` auf `true` oder `false` gesetzt, wenn es sich um Daten eines Familienangehörigen handelte.

Neben den `$queryX`-Variablen wurden noch drei Variablen für Dokumentdatum (`$time`), den Titel des EHR-Dokuments (`$title`) und die URL zum ursprünglichen Dokument (`$link`) gesetzt.

In Listing 5.5 ist ein Ausschnitt der XQuery-Suchanfrage zu sehen, die für ISO/EN-13606-EHR-Extrakte aus der inhaltsbasierten Suche in Listing 5.4 generiert wird. Eine XQuery-Suchanfrage für HL7 CDA oder openEHR könnte nach demselben Schema aufgebaut werden, die jeweiligen Pfade zu den medizinischen Inhalten müssten jedoch an die Vorgaben des jeweiligen Referenzmodells angepasst werden.

### Listing 5.5: XQuery-Suchanfrage

```
1 xquery version '1.0';
2 declare namespace functx = 'http://www.functx.com';
3 declare function functx:distinct-deep ...
4 declare function functx:is-node-in-sequence-deep-equal
5 declare function functx:add-attributes
6 let $time := functx:add-attributes(/*:EHR_EXTRACT/*:time_created, (xs:
    QName('columnName'), xs:QName('columnNumber')), ('Dokumentdatum', '1'))
7 let $title := functx:add-attributes(/*:EHR_EXTRACT/*:all_compositions/*:
    name, (xs:QName('columnName'), xs:QName('columnNumber')), ('Title',
    '2'))
8 let $link := <url columnName='Link' columnNumber='{5}'>URL/LINK</url>
9 let $query1:=for $temp1 in /**[*:archetype_id = 'CEN-EN13606-CLUSTER.
    blood_pressure.v1/at0000']
10 where
11   $temp1/*[*:EHR_EXTRACT/*:time_created/*:time > '2011-04-01T09:54:23']
    and
12   $temp1/*[*:EHR_EXTRACT/*:time_created/*:time < '2011-10-01T09:54:23']
    and
13   $temp1/*[*:archetype_id = 'at0001']/*:value/*:value > 139 and
14   $temp1/*[*:archetype_id = 'at0001']/*:value/*:unit/*:codeValue = 'mm[HG]'
    and
15   $temp1/*[*:archetype_id = 'at0004']/*:value/*:value > 80 and
16   $temp1/*[*:archetype_id = 'at0004']/*:value/*:unit/*:codeValue = 'mm[HG]'
17 return if($temp1/ancestor::*[@:type='ENTRY' and *:
    subject_of_information_category])
18 then functx:add-attributes($temp1, (xs:QName('columnNumber'),xs:QName('
    columnTerm'), xs:QName('otherSubject')), ('3', 'Blutdruck', 'true'))
19 else functx:add-attributes($temp1, (xs:QName('columnNumber'),xs:QName('
    columnTerm'), xs:QName('otherSubject')), ('3', 'Blutdruck', 'false')
    )
20 let $query2...
21 ...
22 let $query3...
23 ...
24 return if(count($query1)=0 and count($query2)=0 and count($query3)=0 )
25 then $query1
26 else <row>{(functx:distinct-deep(<root><row>{($time)}</row><row>{($title)
    }</row><row>{($query1)}</row><row>{($query2)}</row><row>{($query3)}</
    row><row>{($link)}</row></root>//row/*)}</row>
```

Listing 5.6: Ergebnis einer XQuery-Suchanfrage

```
1 <root>
2 <row>...</row>
3 ...
4 <row>
5   <time_created columnName="Dokumentdatum" columnNumber="1" xsi:type="TS
      ">
6     <time >2011-06-03T16:00:00Z</time>
7   </time_created>
8   <name columnName="Title" columnNumber="2">
9     <originalText >Arztbrief vom Internisten</originalText>
10  </name>
11  <items columnNumber="3" columnTerm="Blutdruck" otherSubject="false"
      ...>
12    *** Suchergebnis ***
13  </items>
14  <items columnNumber="4" columnTerm="Laboruntersuchung Glukosestatus"
      otherSubject="false" ...>
15    *** Suchergebnis ***
16  </items>
17  <url columnName="Link" columnNumber="5">URL/LINK</url>
18 </row>
19 </root>
```

Die XQuery wurde vom *Document Crawler* auf die EHR-Dokumente abgesetzt und lieferte die relevanten XML-Segmente zurück.

#### 5.4.6 Gewünschtes Ergebnis der XQuery-Suchanfrage

Das Ergebnis der XQuery-Suchanfrage in den EHR-Dokumenten wurde in einem Ergebnisdokument zusammengefasst, siehe Listing 5.6. Die XQuery-Suchanfrage wurde für jedes EHR-Dokument einzeln ausgeführt und lieferte als Ergebnis pro EHR-Dokument eine einzelne `<row>` zurück. Die `<row>`-Elemente (Zeile 2 und 4–18) wurden im Ergebnisdokument in einem `<root>`-Element zusammengefasst (1–19). Liefert eine Suche kein Ergebnis, wurde kein `<row>`-Element für das jeweilige EHR-Dokument erzeugt.

Jede `<row>` ist nach demselben Schema aufgebaut. Die ersten beiden XML-Elemente der `<row>`, `<time_created>` (Zeile 4–7) und `<name>` (Zeile 8–10), beinhalten den Zeitpunkt, wann das EHR-Dokument erzeugt wurde, sowie den Namen des EHR-Dokuments. Jedes direkte Kind-Element der `<row>` hat ein Attribut `columnNumber` und `columnName`.

Über die `columnNumber` wird jedes Suchergebnis einer Konjunktion der in die disjunktive Normalform umgewandelten Suchanfrage zugewiesen, der `columnName` dient zur leichteren Lesbarkeit des fertigen Dokuments. Die `<time_created>` hat immer die `columnNumber` 1, der `<name>` `columnNumber` 2. Die weiteren direkten Kind-Elemente (Zeile 11–13, Zeile 14–16) beinhalten die XML-Fragmente mit den relevanten Informationen, wie sie im EHR-Dokument abgelegt sind. Diese Elemente können wie im Beispiel-Code `<items>`-Element sein oder `<all_compositions>`-, `<compositions>`-, `<members>`- oder `<content>`-Element, je nachdem ob das Ergebnis der Suchanfrage aus einer COMPOSITION, einer SECTION, einem ENTRY, einem CLUSTER oder einem ELEMENT besteht.

Das Attribut `otherSubject` ist ein Indikator, ob dieser Ausschnitt einem anderen Patienten zugeordnet ist, da diese Information aus dem ausgewählten Ausschnitt nicht mehr ersichtlich sein kann. Neben `otherSubject` könnten auch noch weitere Informationen, die für den Kontext des gewählten Ausschnittes wichtig sind, übergeben werden.

Das letzte direkte Kind-Element einer `<row>` ist immer das `<url>`-Element (Zeile 17) mit der URL zum jeweiligen EHR-Dokument beim jeweiligen GDA. Da die ID eines Dokuments, die innerhalb von IHE-XDS verwendet wird, nicht in den EHR-Dokumenten selbst gespeichert ist und die XQuery aus Performanz-Gründen nicht für jedes Dokument angepasst werden soll, wurde die IHE-XDS-Dokumenten-ID nach dem Ausführen der XQuery in der jeweiligen `<row>` in das `<url>`-Element eingefügt.

#### **5.4.7 Ausführen der Suche in den EHR-Dokumenten**

Die XQuery wurde im nächsten Schritt auf die relevanten Dokumente, die mit der metadatenbasierten Suche in der *Document Registry* gefunden und von der *Document Source* heruntergeladen wurden, angewendet. Die XQueries wurden mit dem Java-XQuery-Prozessor *Saxon* in mehreren Threads gleichzeitig auf die XML-Dokumente angewendet und im *Document Crawler* aufbereitet. Diese Aufbereitung erlaubte es, Wissen über die Struktur der EHR-Dokumente nur an einer Stelle anwenden zu müssen. Der *Document Consumer*, in dem die Ergebnisse dem Benutzer angezeigt wurden, musste kein Wissen über den Aufbau der EHR-Dokumente, der Archetypen und des Referenzmodells haben, sondern bekam eine fertige Struktur übermittelt.

Die mittels *focused retrieval* gefundenen Teile im EHR-Dokument wurden in einer temporären Datenbank im Arbeitsspeicher abgelegt. Die Datenbank bestand aus drei Tabellen, “infoItem”, “InfoItemValues” und “ResultCell”. Die temporäre Datenbank erlaubte es, die Werte nach Dokumentzeitpunkt zu sortieren und mithilfe von einfachen Select-Statements eine XML-Struktur aufzubauen, die unabhängig vom Referenzmodell die Ergebnisse darstellen konnte. Eine Übersicht der Tabellen ist im ER-Diagramm in Abbildung 5.8 zu sehen.

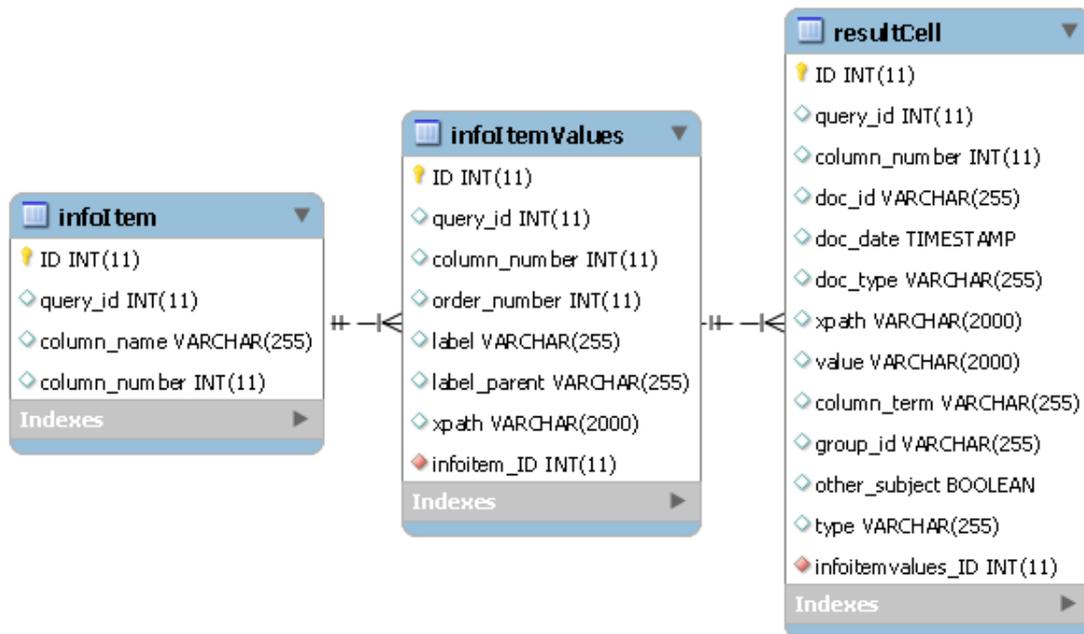


Abbildung 5.8: ER-Diagramm der Tabellen zum Erzeugen der Suchergebnisse.

Die ersten beiden Tabellen (“infoItem”, “infoItemValues”) wurden mithilfe des *Archetype Repository* und den in der Suchanfrage verwendeten Suchbegriffen aufgebaut. Die Tabelle “infoItem” enthält für jede Konjunktion der in die disjunktive Normalform umgewandelten Suchanfrage eine eigene Zeile mit der `columnNumber` und dem `columnName`. In “InfoItemValues” wurden alle elementaren Werte ermittelt und eingetragen. In die “ResultCell”-Tabelle wurden die Ergebnisse der XQuery der einzelnen Dokumente eingefügt und dem jeweiligen “infoItem” und “InfoItemValues” zugeordnet.

In der “resultCell” wurden die Werte der unterschiedlichen Datentypen in einem *String* zusammengefügt. So wurden zum Beispiel beim Datentyp *PhysicalQuantity* die *unit* und der *value* zusammengeführt oder bei einem Datentyp *CodedValue* nur noch der *displayName*

angezeigt.

Die so aufbereiteten Daten dienen als Grundlage für das Erzeugen der im folgenden Abschnitt beschriebenen Tabellendarstellung.

## 5.5 Visualisierung der Suchergebnisse

Die Suchanfragen wurden vom *Document Consumer* an den *Document Crawler* gesendet. Die Suche wurde im *Document Crawler* durchgeführt und die aufbereiteten Ergebnisse wieder an den *Document Consumer* gesendet. Die Ergebnisse konnten im *Document Consumer* ohne Wissen über das zugrundeliegende Referenzmodell dargestellt werden. Ziel war es, eine übersichtliche Darstellung aller Suchergebnisse zu finden, mit der ein schneller Überblick über den Zustand eines Patienten möglich ist.

Suchanfragen können aus einem einzelnen Informationsbedürfnis oder wie bei den vorgefertigten Suchanfragen aus einer Menge verschiedener Informationsbedürfnisse bestehen, die mit UND und ODER miteinander verknüpft sind. Die Informationsbedürfnisse haben ebenfalls eine unterschiedliche Granularität. Sie können sich sowohl auf einzelne Messwerte auf elementarer Ebene (z.B. Körpergewicht) als auch auf Dokument-Ebene (z.B. Laborbefund) beziehen. Die Visualisierung deckt Suchanfragen nach Informationsbedürfnissen beliebiger Zahl und Granularität ab.

Die tabellarische Darstellung der Suchergebnisse ist speziell an die Suchergebnisse der inhaltsbasierten Suche in IHE-XDS angepasst. Es können Ergebnisse von unterschiedlicher Granularität und Umfang angezeigt werden. Die Darstellung basiert auf einer Tabelle, deren Zeilen- und Spalten-Anzahl dynamisch aus der Suchanfrage und den Ergebnissen bestimmt wird. In Abbildung 5.9 ist ein Ausschnitt der zurückgelieferten Suchergebnisse zu sehen.

Im Kopfbereich der Tabelle werden die Spaltenbezeichner sowie eine Zusammenfassung der Suchanfrage angezeigt. Jede Konjunktion der in disjunkte Normalform aufgelösten Suchanfrage wird durch eine Zeile, die eine Zusammenfassung der in der Konjunktion beinhalteten Informationsbedürfnisse angezeigt, dargestellt. In Abbildung 5.10 ist ein Beispiel einer Zusammenfassung mit Werteinschränkungen zu sehen.

Je nach Anzahl der Konjunktionen werden weitere Blöcke dargestellt, die immer nach dem folgenden Muster aufgebaut sind. In der zweiten Spalte "Unterkategorie" werden die

EHR-Arche - Dokumentensuche

Benutzer: Modus Info

Identifikation des Patient

–Angaben zum Patienten : Adam Adamec | geboren am 26.10.1931 | SVNr: 555514 | Geschlecht männlich

Erweiterte Dokumentensuche Präsentation der Ergebnisse **Strukturierte Darstellung der Suchergebnisse**

Kategorie Unterkategorie Suchanfrage: Erstgespräch Zeitraum: 6 Monate

**Diabetes Klassifikation**

	DM Erstdiagnose	01.02.1995	01.02.1995
	Typ 2 Diabetes	Ja	Ja
Diabetes Klassifikation Familienanamnese	Kommentar	Nicht bekannt, ob die Eltern DM ge	Nicht bekannt, ob die Eltern DM ha
		13.09.2011Arztbrief vom Interniste	03.06.2011Arztbrief vom Interniste

**Allgemeinzustand**

	Claudicatio intermittens		Ja
Stuhlgang	Regelmäßiger Stuhlgang	Ja	Ja
	Diarrhoe	Nein	
	Obstipation	Ja	Ja
Nausea	Nausea	Ja	Ja
	Wie oft?	hat sich gebessert, nur manchmal	Gelegentlich, nicht mehr so oft.
Nykturie	Nykturie	Ja	Ja
	Wie oft pro Nacht?	2 mal pro Nacht	2 mal
		13.09.2011Arztbrief vom Interniste	03.06.2011Arztbrief vom Interniste

**Lebensstil**

Abbildung 5.9: Aufbau der Rückgabetable (aus [21]).

Beschriftungen der elementaren Werte angezeigt (1a und 1b). In der ersten Spalte “Kategorie” werden die Beschriftungen der darüberliegenden Klassen bis zum gemeinsamen Elternknoten angezeigt (2). Gibt es keine darüberliegenden Klassen, wird keine Beschriftung angezeigt. Ab der dritten Spalte werden die Suchergebnisse zu den jeweiligen Klassen aus den unterschiedlichen Dokumenten angezeigt (3). Die Dokumente sind in chronologischer Reihenfolge dargestellt, wobei sich die Werte des neuesten Dokuments ganz links und die Werte des ältesten ganz rechts befinden. Alle Werte, die aus dem gleichen Dokument stammen, werden von einem dünnen Rahmen umgeben, in der Fußzeile ist jeweils ein Link zum Originaldokument sowie der Name und das Datum des dazugehörigen Dokuments dargestellt (4). Innerhalb eines Dokuments werden mehrfach dokumentierte Werte in der Reihenfolge, in der sie dokumentiert wurden dargestellt, das Dokument wird dabei in mehrere Spalten aufgeteilt abgebildet. Ist ein Text zu breit für die Spalte, wird ein kleines graues Dreieck am unteren rechten Rand der Zelle eingeblendet. Über einen eingeblendeten Hilfetext (“Tooltip”)

wird der gesamte Inhalt der Zelle angezeigt. Die Breite der Spalte kann zusätzlich im Kopfbereich der Tabelle verändert werden.

Kategorie	Unterkategorie	Suchanfrage: Kurzabfrage Blutdruck	Zeitraum: 6 Monate
▲ Systolisch > 139 und Diastolisch > 80			
Blutdruck	Zeitpunkt	13.09.2011 08:00	03.06.2011 08:00
	Art der Blutdruckmessung	Blutdruckmessung im Sitzen	Blutdruckmessung im Sitzen
	Systolisch	150 mm[HG]	145 mm[HG]
	Diastolisch	95 mm[HG]	85 mm[HG]
		13.09.2011Arztbrief vom Internisten	03.06.2011Arztbrief vom Internisten
▲ Analyse = HbA1c (glykosyliertes Hämoglobin A1C)			
	Analyse	HbA1c (glykosyliertes Hämoglobin A	HbA1c (glykosyliertes Hämoglobin A
	Ergebnis	10.0 %	10.0 %
	Referenzbereich	4.4 % - 6.0 %	4.4 % - 6.0 %
	Interpretation	H	H
		13.09.2011Arztbrief vom Internisten	13.09.2011 Laborbefund

Abbildung 5.10: Aufbau der Rückgabetable mit Werteinschränkungen.

## 5.6 Evaluierung und Laufzeiten

Die vorgestellte inhaltsbasierte Suche in einer IHE-XDS-Umgebung wurde im Rahmen des EHR-Arche-Projekts von sieben Internisten mit einer Spezialisierung auf Diabetes mellitus evaluiert. Die Internisten/innen kamen aus dem Allgemeinen Krankenhaus der Stadt Wien, dem Landeskrankenhaus Innsbruck, dem Bezirkskrankenhaus Hall in Tirol, dem Landeskrankenhaus Natters, dem Krankenhaus St. Vinzenz in Zams sowie aus dem niedergelassenen Bereich. Die Evaluierung wurde von einer Medizinerin durchgeführt. Der genaue Ablauf der Evaluierung ist in [21] zusammengefasst. Im Folgenden werden die Ergebnisse der Evaluierung die Suche betreffend beschrieben.

Um die Laufzeiten der Suche zu analysieren, wurden sechs vorgefertigte Suchanfragen erzeugt, die auf die daraus generierte metadatenbasierte Suchanfrage jeweils dieselben 48 EHR-Dokumente zurücklieferten. Es wurde ein PC mit Intel(R)-Core(TM)2-Quad-Prozessor mit 2,66 GHz, 8 GB RAM und Windows 7 64 Bit SP1 als Betriebssystem zum Testen verwendet. Die sechs Suchanfragen sind wie im Folgenden beschrieben aufgebaut:

- ein Suchbegriff
- 10 unterschiedliche Suchbegriffe, die mit ODER verknüpft sind
- 100 unterschiedliche Suchbegriffe, die mit ODER verknüpft sind
- 2 unterschiedliche Suchbegriffe, die mit UND verknüpft sind
- 5 unterschiedliche UND-Suchbegriffspaare, die jeweils mit ODER verknüpft sind
- 90 unterschiedliche Suchbegriffe und 5 UND-Suchbegriffspaare, die jeweils mit ODER verknüpft sind

Alle sechs Suchanfragen wurden pro Testlauf in unterschiedlicher Reihenfolge ausgeführt, danach wurde der *Document Crawler* neu gestartet. In Tabelle 5.1 sind die Zeiten der sechs Testläufe dargestellt, wobei die erste durchgeführte Suche, nachdem der *Document Crawler* neu gestartet wurde, mit grau hinterlegt ist. Aus Tabelle 5.1 ist ersichtlich, dass nach einem Neustart die Suche viel länger dauert. Dies hängt damit zusammen, dass das *Archetype Repository* und die temporäre Datenbank zur Erzeugung der Ergebnistabelle nach jedem Neustart neu initialisiert werden müssen. Da dies nur bei der ersten Suchanfrage nach einem Neustart gemacht werden muss, wurden die als Erstes durchgeführten Suchanfragen nicht in die Berechnungen des Mittelwertes, der Standardabweichung (SD) und der Summe in Tabelle 5.1 einbezogen. Bei der als Erstes durchgeführten Suchanfrage werden zusätzlich die EHR-Dokumente vom *Document Repository* geladen. Das Herunterladen der Dokumente ist stark abhängig von der zugrundeliegenden Netzwerkinfrastruktur und nimmt rund 50% der Zeit in Anspruch. Um die Suchzeiten zu verringern, wird ein Dokument nur beim ersten Auffinden heruntergeladen und für die weitere Verwendung während derselben Sitzung im *Document Crawler* zwischengespeichert. Da bei jeder durchgeführten Suchanfrage des Testlaufs dieselben 48 EHR-Dokumente durchsucht wurden, mussten bei den folgenden Suchanfragen (alle nicht grau hinterlegten Werte in Tabelle 5.1) in einem Testlauf keine EHR-Dokumente mehr vom *Document Repository* geladen werden.

Im Verarbeitungsschritt “Metasuche erzeugen” wird die XML-basierte Suchanfrage in eine Java-Instanz konvertiert und gleichzeitig die für die metadatenbasierte Suchanfrage benötigten Metadaten wie zum Beispiel Patienten-ID herausgeholt und die Dokumenttypen mithilfe des *Archetype Repository* berechnet. Im Schritt “XQuery erzeugen” wird aus der Java Instanz der

Suchanfrage mithilfe des *Archetype Repository* die XQuery generiert. In diesem Schritt wird auch die Struktur der Rückgabetable in der temporären Datenbank angelegt. Im Schritt "XQuery ausführen" werden die EHR-Dokumente vom *Document Repository* bzw. falls sie in dieser Session schon geladen wurden aus dem Cache geladen und die XQuery wird angewendet. Die Ergebnisse werden in der temporären Datenbank abgelegt. Im Schritt "Ergebnisse darstellen" werden die Suchergebnisse aus der temporären Datenbank geladen und die Ergebnistabelle erstellt.

Die Dauer der analysierten Verarbeitungsschritte ist abhängig von der Anzahl der verwendeten Suchbegriffe in der inhaltsbasierten Suchanfrage. Das Erzeugen der metadatenbasierten Suchanfrage dauerte jedoch immer kürzer als 0,13 Sekunden und das Erzeugen der XQuery nie mehr als 1,35 Sekunden. Die benötigte Zeit zum Ausführen der XQuery ist neben der Anzahl der verwendeten Suchbegriffe auch von der Anzahl der zu durchsuchenden Dokumente abhängig. Die Zeit zum Erstellen der Ergebnistabelle hängt zusätzlich von der Anzahl der durch die XQuery gefundenen Werte ab.

Der *Document Consumer* und die umgesetzte Suche wurden im Rahmen der Evaluierung als ausreichend schnell und stabil empfunden. Die inhaltsbasierte Suche mit ISO/EN-13606-EHR-Extrakten wurde dabei mit einer metadatenbasierten Suche mit inhaltlich identen PDF-Dokumenten verglichen. Die inhaltsbasierte Suche wurde als einfach und selbsterklärend empfunden, sie sei intuitiver und schneller als die metadatenbasierte Suche. Die vorgefertigten Suchanfragen wurden ebenfalls als hilfreich empfunden, um einen schnellen Überblick über einen Patienten zu bekommen. Die Informationsüberflutung war mit der inhaltsbasierten Suche besser zu bewältigen als mit der herkömmlichen metadatenbasierten Suche [21].

Tabelle 5.1: Dauer (in Sekunden) der unterschiedlichen Verarbeitungsschritte mit Mittelwert und Standardabweichung (SD) der sechs Testläufe. Die grau hinterlegten Zellen stammen vom jeweils ersten Suchdurchlauf nach einem Neustart des *Document Crawler*. Sie werden in der Berechnung des Mittelwerts und der Standardabweichung nicht berücksichtigt.

Testlauf	1	2	3	4	5	6	Mittelwert	SD
<b>1 Suchbegriff</b>								
Metasuche erzeugen	2,100	0,002	0,000	0,000	0,000	0,000	0,000	0,0008
XQuery erzeugen	2,498	0,004	0,015	0,016	0,016	0,000	0,010	0,0076
XQuery ausführen	12,644	0,640	0,638	0,734	0,469	0,469	0,590	0,1170
Ergebnis darstellen	0,390	0,031	0,016	0,032	0,015	0,031	0,025	0,0086
Summe	17,632	0,677	0,669	0,782	0,500	0,500	0,626	
<b>10 Suchbegriffe mit ODER verknüpft</b>								
Metasuche erzeugen	0,033	0,000	2,044	0,000	0,015	0,000	0,010	0,0146
XQuery erzeugen	0,156	0,047	3,855	0,031	0,062	0,094	0,078	0,0494
XQuery ausführen	2,921	1,113	13,046	0,906	1,514	1,155	1,522	0,8122
Ergebnis darstellen	0,686	0,156	1,030	0,156	0,237	0,438	0,335	0,2277
Summe	3,796	1,316	19,975	1,093	1,828	1,687	1,944	
<b>100 Suchbegriffe mit ODER verknüpft</b>								
Metasuche erzeugen	0,107	0,028	0,135	2,169	0,117	0,079	0,093	0,0417
XQuery erzeugen	0,796	0,624	0,826	5,183	1,342	0,983	0,914	0,2709
XQuery ausführen	8,335	7,216	8,130	18,988	8,667	8,145	8,099	0,5387
Ergebnis darstellen	1,264	0,999	1,359	2,113	1,842	1,467	1,386	0,3081
Summe	10,502	8,867	10,450	28,453	11,968	10,674	10,492	
<b>2 Suchbegriffe mit UND verknüpft</b>								
Metasuche erzeugen	0,000	0,000	0,000	0,015	2,184	0,015	0,006	0,0086
XQuery erzeugen	0,016	0,016	0,000	0,016	3,714	0,016	0,013	0,0092
XQuery ausführen	0,954	0,594	0,469	0,921	11,643	0,937	0,775	0,2334
Ergebnis darstellen	0,016	0,015	0,016	0,016	0,282	0,078	0,028	0,0005
Summe	0,986	0,625	0,485	0,968	17,823	1,046	0,822	
<b>5 UND-Suchbegriffspaare mit ODER verknüpft</b>								
Metasuche erzeugen	0,063	0,079	0,063	0,056	0,039	2,316	0,060	0,0144
XQuery erzeugen	0,140	0,472	0,156	0,102	0,054	3,871	0,185	0,1652
XQuery ausführen	2,342	1,772	1,467	1,099	0,863	12,336	1,509	0,5809
Ergebnis darstellen	0,031	0,047	0,016	0,026	0,017	0,344	0,027	0,0126
Summe	2,576	2,370	1,702	1,283	0,973	18,867	1,781	
<b>90 Suchbegriffe und 5 UND-Suchbegriffspaare mit ODER verknüpft</b>								
Metasuche erzeugen	0,100	2,528	0,049	0,127	0,162	0,086	0,105	0,0425
XQuery erzeugen	0,608	1,843	0,530	0,655	0,702	0,583	0,616	0,0660
XQuery ausführen	7,572	17,450	6,616	8,022	7,869	6,878	7,391	0,6170
Ergebnis darstellen	1,000	2,293	0,875	1,017	0,968	0,924	0,957	0,0578
Summe	9,280	24,114	8,070	9,821	9,701	8,471	9,069	

## Diskussion

Die vorgestellte Methode zum Suchen in Shared-EHR-Systemen mit zentraler Metadatenkomponente wurde in einem Prototypen implementiert. Im Folgenden werden alternative Umsetzungen zu den angewendeten Methoden analysiert. Weiters werden die Erzeugung von Archetypen, die Validierung und die Erzeugung der Testdokumente und die Anwendbarkeit der vorgestellten Methode auf die in Österreich geplante ELGA diskutiert.

### 6.1 Diskussion der umgesetzten inhaltsbasierten Suche

Die Suche über domänenunspezifische Metadaten verringert die Missbrauchsgefahr der Daten, da diese nicht zentral gespeichert werden, die Suchmöglichkeiten werden dadurch jedoch stark eingeschränkt. Alternativ können die EHR-Dokumente transformiert und in ein lokales GDA-System importiert werden oder direkt im XML-Format lokal indiziert und durchsucht werden. In beiden Fällen ist jedoch Wissen über die Struktur der EHR-Dokumente notwendig. Nach einem Import der Daten kann die im GDA-System angebotene Suche verwendet werden, der Import setzt aber Wissen voraus. Im zweiten Fall ist das Wissen für das Implementieren der XML-Suche notwendig. Kommen neue Dokumenttypen dazu, müssen neue Importfilter bzw. Suchalgorithmen entwickelt werden. Durch eine zentrale Wissensbasis, in der neue Dokumenttypen eingearbeitet werden, verringert sich der Aufwand für das Softwaresystem des GDAs.

Der *Document Crawler* wurde in der vorliegenden Arbeit als zentraler Akteur umgesetzt, alternativ könnte die Funktionalität auch direkt bei den lokalen EHR-Systemen der GDAs realisiert werden. Bei der lokalen Umsetzung direkt im EHR-System der GDAs müssten alle EHR-Dokumente vor der inhaltsbasierten Suche heruntergeladen werden, wodurch für Patienten im Log der Eindruck entsteht, dass die GDAs immer alle EHR-Dokumente ansehen. Bei der zentralen Lösung werden im Gegensatz dazu nur diejenigen EHR-Dokumente zum *Document Consumer* gesendet, deren Inhalte die GDAs angezeigt bekommen.

Die vorgestellte Arbeit konzentriert sich auf die Suche in Shared-EHR-Systemen mit zentraler Metadatenkomponente. Andere Architekturen von EHR-Systemen, wie etwa die unterschiedlichen Implementierungen von zentraler Speicherung, werden nicht behandelt. Bei einer zentralen Speicherung ist das Erstellen eines zentralen Index über alle EHR-Inhalte möglich, das Durchsuchen der einzelnen EHR-Dokumente kann somit entfallen. Soll für die Erstellung des Index die Struktur der zugrundeliegenden EHR-Dokumente berücksichtigt werden, ist das Wissen über die Struktur notwendig. Das eingeführte *Archetype Repository* könnte für die Erstellung des zentralen Index verwendet werden.

Die verwendete thesaurusbasierte Suche erleichtert die Verarbeitung der Suchanfrage, da nur bekannte Suchbegriffe verwendet werden. Archetypen beschreiben genau den Inhalt, der in einem Dokument erfasst wird. Dies erlaubt es, ein eingeschränktes Vokabular aus den Archetypen zu generieren, das für eine Suche verwendet werden kann. Durch die zu jedem Suchbegriff hinterlegten Informationen im *Archetype Repository* ist es direkt möglich, eine metadatenbasierte Suchanfrage zu generieren und die Suchbegriffe den einzelnen Dokumenten und Dokumentteilen zuzuweisen. Bei einer Begriff-Suche oder Freitextsuche sind im Gegensatz zu einer thesaurusbasierten Suche komplexere Suchanfragen möglich. Für die Verwendung mit dem *Archetype Repository* müssten die jeweiligen Begriffe auf die jeweiligen Dokumente und Dokumentteile gemappt werden. Dies setzt das Finden von orthographisch ähnlichen Wörtern und Synonymen etc. voraus. Im Rahmen des *Unified Medical Language System* (UMLS) [46] werden Tools für das Parsen von Freitexten und das Mappen auf existierende Terminologien zur Verfügung gestellt. Diese sind jedoch hauptsächlich für den englischen Sprachgebrauch optimiert.

Ein weiterer Vorteil der thesaurusbasierten Suche besteht darin, dass genau jene Begriffe, die in den Dokumenten auch verwendet werden, als Suchbegriffe angeboten werden. Dem GDA wird

keine feinere Granularität der Daten suggeriert als in den EHR-Dokumenten vorhanden ist. Sind die zugrundeliegenden EHR-Dokumente nur sehr grob strukturiert, werden nur wenige Begriffe in der Suche angeboten. Ist eine feinere Suche gefordert, müssten die jeweiligen Inhalte in den EHR-Dokumenten mit einer feineren Struktur modelliert werden. Es ist ebenfalls möglich, zu den jeweiligen Begriffen manuell weitere Begriffe im Thesaurus zu spezifizieren. Ein Beispiel hierfür ist der Begriff "Familienanamnese", der nicht weiter unterteilt ist. Ist bekannt, dass unter der Familienanamnese die "Rauchgewohnheiten" und der "Blutdruck" angegeben werden können, könnten diese Begriffe manuell zum Thesaurus hinzugefügt werden. In der hier vorgestellten Methode würde aber trotzdem nur die Container-Klasse "Familienanamnese" zurückgeliefert werden, da keine Freitextsuche implementiert wurde.

Im beschriebenen Suchverfahren werden nur jene Elemente in den EHR-Dokumenten gesucht, die im Archetyp beschrieben sind, Freitextfelder werden nicht durchsucht. Die Begriffe, die in der Suchanfrage angegeben werden können, bestehen nur aus den Begriffen, die im Archetyp beschrieben sind. Hat ein Freitextfeld den Namen "Familienanamnese", wird als Suchbegriff nur "Familienanamnese" angeboten. Werden Blutdruckprobleme eines Familienmitglieds in diesem Feld als Freitext dokumentiert und "Blutdruck" als Suchanfrage abgesetzt, werden die Blutdruckprobleme in der jetzigen Implementierung nicht gefunden. "Blutdruck" erscheint als Begriff des Thesaurus, da der Blutdruck des Patienten an einer anderen Stelle in einem Archetyp spezifiziert wurde. Eine Freitextsuche in allen Freitextfeldern könnte als zusätzliche Funktion angeboten werden, um Inhalte, die durch die thesaurusbasierte Suche nicht gefunden wurden, anzuzeigen. Dies wäre vor allem für EHR-Dokumente mit einem geringen Strukturierungsgrad als zusätzliche Erweiterung interessant.

Die Suche in den EHR-Dokumenten wird in der vorliegenden Arbeit mittels XQuery und XPath durchgeführt. *XQuery full text* [161] erlaubt es, zusätzlich zu den XPath-Anweisungen auch Freitextsuchen in XML-Dokumenten auszuführen. XPath ist nicht dafür ausgelegt, Freitextsuchen auszuführen. Mit *Narrowed Extended XPath* (NEXI) [162] wird zusätzlich zu der *contain*-Funktion in XPath, die es erlaubt, nach einem bestimmten Text zu suchen, die *about*-Funktion eingeführt. Mit dieser Funktion kann man Anfragen formulieren, in denen nicht der genaue Wortlaut der Suchanfrage im Dokument enthalten sein muss, um als Ergebnis zurückgeliefert zu werden. In Kombination mit einem Rankingmechanismus kann so eine Freitextsuche in XML umgesetzt werden.

Um den Kontext einzelner EHR-Inhalte im Suchergebnis nicht zu verlieren, wurde beim Erstellen der Suche darauf geachtet, dass Informationen zum Kontext (i.e. Inhalte darüberliegender Elemente) erhalten bleiben. Wurde ein Wert, der mittels der ELEMENT-Klasse oder CLUSTER-Klasse abgebildet ist, gesucht, wurde in der darüberliegenden ENTRY-Klasse überprüft, ob es sich um Information zum selben Patienten wie im EHR handelt oder um ein Familienmitglied etc. In der Ergebnisdarstellung werden Daten, die nicht vom Patienten stammen, grau hinterlegt angezeigt. Weitere kontextabhängige Informationen (z.B. ältere/neuere Version vorhanden, keine gesicherte Information etc.) wurden nicht implementiert.

Umrechnungen zwischen Einheiten wurden nicht implementiert. Können Werte mit unterschiedlichen Einheiten dokumentiert werden, werden beim Erstellen der Suchanfrage alle möglichen Einheiten angezeigt. Um Einschränkungen über mehrere Einheiten zu machen, müssen diese Einschränkungen mit ODER verknüpft werden und für jede mögliche Einheit dieselbe Einschränkung gemacht werden, da nur dann die Kombination aus jeweils Einheit und Wert miteinander ausgewertet werden kann.

Die erzeugten XQuery-Suchanfragen und die darin vorkommenden XPath-Anweisungen sind speziell an die Struktur der zugrundeliegenden EHR-Dokumente angepasst. Durch das automatische Erzeugen der XQuery-Suchanfragen muss beim Erstellen der Suche kein Vorwissen über die Struktur vorhanden sein. Das Berücksichtigen der Struktur erlaubt es, explizit auf einzelne Elemente zuzugreifen. Variiert die Struktur, was bei archetypbasierten EHR-Dokumenten eigentlich nicht vorkommen darf, werden nur die Elemente zurückgeliefert, die genau den Vorgaben des Archetyps entsprechen.

In [104] wird eine Vision einer zukünftigen Suche mit existierenden Technologien beschrieben. Der erste Schritt der Vision, die Extraktion relevanter Ergebnisse, wurde in der vorliegenden Arbeit umgesetzt. Die Suchanfragen können mittels der thesaurusbasierten Suche leicht erzeugt werden und werden im Hintergrund zu komplexen Suchen, die die Struktur und die Terminologien berücksichtigen, zusammengefasst. Weitere Informationsquellen wie Behandlungspläne und klinische Guidelines könnten, sofern diese in strukturierter Form vorliegen, über das *Archetype Repository* und die *Termbindings* in Archetypen integriert werden, um den GDAs eine zusätzliche Entscheidungsunterstützung anzubieten. Mit Archetypen beschriebene EHR-Dokumente haben das Potenzial, einige dieser Visionen

möglich zu machen.

Im EHR-Arche Projekt wurden ISO/EN-13606-Archetypen verwendet. Manche Programmteile wurden jedoch auch mit openEHR und CDA-Archetypen getestet. Es ergaben sich dabei keine Probleme, die das Anwenden der vorgestellten Methode auf den Standard ISO/EN 13606 beschränken.

### **6.1.1 Vergleich mit existierenden Suchansätzen**

*MorphoSaurus* [107] ist eine Dokumentensuchmaschine, die in lokale EHR-Systeme integriert wird. Die in der vorliegenden Arbeit vorgestellte Methode zur inhaltsbasierten Suche legt den Fokus auf Shared-EHR-Systeme, bei denen kein zentraler Index aufgebaut werden kann.

Im *RAVEL*-Projekt [108] werden EHR-Daten semantisch indiziert und natürlichsprachliche Suchanfragen unterstützt. Im Gegensatz zum Index in der vorliegenden Arbeit, basiert der Index im *RAVEL*-Projekt auf den Inhalten in den EHR-Dokumenten. Durch den archetypbasierten Index in der vorliegenden Arbeit können die jeweiligen Teile eines EHR-Dokuments direkt an der jeweiligen Stelle im EHR-Dokument extrahiert werden. Der Index im *RAVEL*-Projekt kann die Werte direkt aus dem Index bereitstellen, was jedoch einen geringeren Datenschutz bewirkt, da die Werte zusätzlich in einem zentralen Index abgelegt werden. Die im *RAVEL*-Projekt entwickelten Methoden zur Visualisierung, aber auch zum semantischen Indizieren, können jedoch für ein Optimieren der Ergebnispräsentation und der Erzeugung der Wissensbasis herangezogen werden.

Ähnlich zu [109] werden in der vorliegenden Arbeit Technologien des Semantic Web (i.e. *RDF-Triple*) für die Suche verwendet. Die *RDF-Triple* werden jedoch aus den EHR-Dokumenten generiert und es wird ein zentraler Index aufgebaut. Der Index wird bei der Suche in einer IHE-XDS-Umgebung verwendet, um eine inhaltsbasierte Suche anbieten zu können. In [110] wird der zentrale Index mittels einer XML-Datenbank umgesetzt.

Die in der vorliegenden Arbeit präsentierte inhaltsbasierte Suche verwendet einen ähnlichen Ansatz wie [111], um die metadatenbasierte Suche zu optimieren. Die dazu benötigte Wissensbasis wird jedoch aus Archetypen generiert und nicht aus EHR-Dokumenten. Zusätzlich werden im Gegensatz zu [111] nicht ganze Dokumente, sondern nur die gesuchten Teile eines EHR-Dokuments zurückgeliefert.

In *XOntoRank* [112] werden standardisierte Ontologien wie Snomed-CT verwendet, um in CDA-Dokumenten zu suchen. Die in der vorliegenden Arbeit verwendeten Informationsbedürfnisse sind nicht auf eine standardisierte Ontologie gemappt. Für ein System im Routinebetrieb ist ein solches Mapping eine wichtige Ergänzung, um die in den EHR-Dokumenten und den Archetypen verwendeten Codes optimal nutzen zu können.

Die ebenfalls auf Archetypen basierenden Systeme *Query Builder* [114], *LinkEHR* [61] und *ByMedConnect* [35, 36] erlauben es, Suchen basierend auf Archetypen durchzuführen. Keines dieser Systeme bietet jedoch eine Integration mit IHE-XDS an.

Im Rahmen von IHE werden ebenfalls Profile für eine inhaltsbasierte Suche entwickelt (z.B. *Query for Existing Data* (QED) [117], bzw. *On-Demand Documents* [122]). Im Profil *On-Demand Documents* werden jedoch nur Dokumente einer einzelnen *Document Source* zusammengefasst. Dies müsste in allen *Document Sources* implementiert werden. Der in der vorliegenden Arbeit präsentierte Ansatz kann in eine existierende IHE-XDS-Architektur integriert werden. Neben dem Zur-Verfügung-Stellen von standardisierten Dokumenten werden keine zusätzlichen Anforderungen an existierende *Document Sources* gestellt.

## **6.1.2 Alternative Umsetzungen der inhaltsbasierten Suchanfrage**

Die in dieser Arbeit verwendeten inhaltsbasierten Suchanfragen wurden aus den erhobenen Informationsbedürfnissen und boolschen Verknüpfungen zusammengesetzt. Dies erlaubte es, referenzmodellübergreifende Suchanfragen zu formulieren, und deckte die Anforderungen, die an das EHR-Arche-Projekt gestellt wurden, ab. Für zukünftige Anwendungen sollten weitere Anfragesprachen analysiert und als Alternativen in Betracht gezogen werden, um ebenfalls auf existierende Tools zurückgreifen zu können.

In [133] wird die boolsche Suche als nicht brauchbar zur Verhinderung von Informationsüberflutung angesehen, da kein Ranking der Ergebnisse möglich ist. Die Wichtigkeit des Rankings bei Suchen im Internet steht außer Frage. Werden jedoch nur EHR-Dokumente eines einzelnen Patienten durchsucht, deren Anzahl um einige Größenordnungen kleiner ist als bei Suchen im Web, kann ein zeitliches Ranken der Ergebnisse ausreichend sein, um Informationsüberflutung zu minimieren. Durch den hohen Strukturierungsgrad der verwendeten EHR-Dokumente können die Ergebnisse der Suche übersichtlich dargestellt werden. Dies ist ein wichtiger Schritt, um Informationsüberflutung zu

vermeiden. Die Ergebnisse der Evaluierung der Suche (siehe [21]) zeigen, dass die verwendete boolesche Suche in dem getesteten Anwendungsfall Informationsüberflutung vermeiden kann.

Der *Query Builder* von Ocean Informatics [114] erlaubt es, mithilfe der Pfade zu Archetyp-Knoten Suchanfragen in der *Archetype Query Language* zu erzeugen. Die *Archetype Query Language*-Spezifikation wurde auf der openEHR-Seite veröffentlicht, es existieren jedoch im Gegensatz zur ADL keine frei zugänglichen Implementierungen oder APIs dafür. Der Vorteil der *Archetype Query Language*, dass sie unabhängig von der Art der Persistierung der Dokumente (i.e. XML, Datenbank, Objekte etc.) verwendet werden kann, war für die vorliegende Arbeit nicht relevant, da EHR-Dokumente in IHE-XDS generell als XML-Dokumente vorliegen. Dies erlaubt es, die Suche mittels XQuery durchzuführen. XQuery ist ein offizieller Standard und es existieren frei zugängliche Tools. Die XQuery wird in der vorliegenden Arbeit nicht direkt vom Benutzer erstellt, sondern aus einer eigenen Abfragesprache generiert.

Suchanfragen wurden in der vorliegenden Arbeit immer nur auf ein einzelnes Dokument angewendet, eine dokumentübergreifende Suche wurde nicht berücksichtigt. Werden in einer Suchanfrage zwei Begriffe mit UND verknüpft, müssen beide Ausprägungen im selben Dokument vorkommen. Wird zum Beispiel nach "Operation" UND "Laborwert" gesucht, wobei "Operation" in Dokument A und der "Laborwert" in einem Dokument B vom selben Tag gespeichert sind, liefert die Suche keine Ergebnisse. Das Finden temporaler Zusammenhänge zwischen Dokumenten war nicht Fokus der vorliegenden Arbeit. Die umgesetzte Suche mit XQuery kann nicht direkt für eine dokumentübergreifende Suche verwendet werden. Die einzelnen Teile der UND-Verknüpfung könnten mit XQuery aus den Dokumenten geladen werden, die dokumentübergreifenden Beziehungen müssten dann separat, unabhängig von XQuery, überprüft werden.

Neben den verwendeten Begriffen kann auch die hierarchische Struktur des Archetyps bei der Erzeugung der Suchanfrage berücksichtigt werden. So macht es einen Unterschied, ob eine "Blutdruckmessung" bei einer "Familienanamnese" gesucht werden soll, in der Dokumentation vom jeweiligen Patienten oder prinzipiell alle "Blutdruckmessungen" angezeigt werden sollen. Die Hierarchie der Informationsbedürfnisse wurde beim Erstellen von komplexen Suchanfragen noch nicht berücksichtigt. Beim Erstellen der Suchanfrage könnten über ein Anzeigen der darüberliegenden Hierarchie von Informationsbedürfnissen die gewünschten

Informationsbedürfnisse zusätzlich eingeschränkt werden. Wird der “Blutdruck” unter “Familienanamnese” ausgewählt, werden nur diese “Blutdruck”-Werte zurückgeliefert. In der momentanen Implementierung kann diese Unterscheidung nicht automatisch gemacht werden, sondern muss manuell in der Suchanfrage hinzugefügt werden.

Beim Erzeugen von Suchanfragen muss beachtet werden, dass sich die genaue Hierarchie der Daten abhängig vom zugrundeliegenden Referenzmodell unterscheiden kann. Werden in der Suchanfrage hierarchische Strukturen vorgegeben, können Instanzen eines Referenzmodells von der Suche dadurch ausgeschlossen werden. Dieser Umstand muss beim Erzeugen der Informationsbedürfnisse und deren Hierarchie ebenfalls berücksichtigt werden. Die in der vorliegenden Arbeit verwendeten Suchbegriffe sind ident mit den im Rahmen des EHR-Arche-Projekts erhobenen Informationsbedürfnissen der GDAs im Rahmen der Diabetes-mellitus-Behandlung und die Hierarchie wird direkt aus den Archetypen übernommen. Weicht die verwendete Hierarchie der Suchbegriffe stark von der Hierarchie der dokumentierten Inhalte ab, muss dieser Ansatz überarbeitet werden, da es zu Widersprüchen in den beiden Hierarchien kommen kann.

Die in der thesaurusbasierten Suche angebotenen Suchbegriffe stammen von einer Informationsbedarfsanalyse, die speziell auf die Bedürfnisse der Ärzte abgestimmt war. Es ist nicht geklärt, ob die thesaurusbasierte Suche für semistrukturierte Dokumente, wie sie etwa bei der CDA vorkommen, genauso zufriedenstellend ist, da in diesem Fall weniger Suchbegriffe bzw. nur Suchbegriffe mit einer gröberen Granularität angeboten werden können.

### **6.1.3 Optimieren der inhaltsbasierten Suche**

Die vorgestellte Methode sollte die potenzielle Nutzung von Archetypen bei der inhaltsbasierten Suche in Shared-EHR-Systemen zeigen. Die Performance spielte dabei nur eine nebensächliche Rolle. Um die Suche mittels XQuery zu beschleunigen, wurden die Dokumente nur bei der ersten Anfrage vom *Document Repository* geladen und dann zwischengespeichert. Es wurde kein Index erstellt oder andere Optimierungen durchgeführt. Im Folgenden werden einige Mechanismen besprochen, die zur Optimierung von zukünftigen Systemen angewendet werden können. Neben Optimierungen der XQuery-Suche werden noch andere alternative Suchen angesprochen.

In XML-Datenbanken werden XQuery-Suchanfragen, bevor sie abgesetzt werden,

umgeschrieben und optimiert, um die Performance zu steigern [163]. Dabei werden Mechanismen, die aus der Optimierung von SQL-Anfragen bekannt sind wie etwa der *Pushdown*-Mechanismus, auf XQuery-Suchanfragen angewendet. Die Optimierung der XQuery-Suchanfragen wurde nicht angewendet, da das Erzeugen der optimierten XQuery ebenfalls Zeit in Anspruch nimmt. Für große XQuery-Suchanfragen, wie sie etwa bei den vorgefertigten Suchanfragen verwendet werden, ist eine zukünftige Optimierung sinnvoll. Die optimierten XQuerys sollten dazu aber zusätzlich zwischengespeichert werden. Dies ist nur bedingt möglich, da diese bei Änderungen in der Wissensbasis immer neu generiert werden müssen.

Die erzeugte XQuery-Suchanfrage geht nur wenig auf die Anforderungen der Visualisierung ein. Die aus der XQuery erzeugte Struktur wird in einem weiteren Schritt in das Ausgabeformat transformiert. Beim Erstellen der XQuery könnte noch detaillierter auf die gewünschte Struktur zur Visualisierung eingegangen werden, um ein weiteres Parsen der XQuery-Ergebnisse zu vermeiden.

Um Ressourcen bei der Erzeugung der XQuery einzusparen, könnte die Erstellung der XQuery auf den Client (i.e. den Rechner des GDA) ausgelagert werden. Beim Erstellen der Suchanfrage könnten direkt relevante Knoten in der Wissensbasis gesucht werden. In der Zeit, in der der Benutzer die Suchanfrage nochmals anschaut, den Zeitbereich auswählt oder mit der Maus zum Suchknopf geht, könnte im Hintergrund die XQuery erzeugt werden.

Beim Ausführen einer XQuery kann das Wissen über die Struktur (i.e. ein XML-Schema) die Ausführungszeit verkürzen, führt aber auch zu einem zusätzlichen Overhead. Es wurde die freie Version von *Saxon* verwendet. Diese Version unterstützt im Gegensatz zur kommerziellen Version die Berücksichtigung von XML-Schema bei der Ausführung von XQuery nicht. Die Vor- und Nachteile müssen abhängig von den zu durchsuchenden EHR-Dokumenten und den jeweiligen XQuerys analysiert werden.

Durch die Verwendung von XML beim Datenaustausch von Gesundheitsdaten kann es, gegenüber der Übertragung von Binärdaten, zu einem erheblich größeren Datenaufkommen kommen, was in weiterer Folge zu einer längeren Übertragungsdauer führt. Die meiste Zeit bei der Suche nach EHR-Dokumenten nimmt der Download der Dokumente vom *Document Repository* in Anspruch. Um diesem Problem entgegenzuwirken, können die XML-Dokumente wie in [74] beschrieben komprimiert werden. Die Komprimierung von Text-Dokumenten

erzielt dabei sehr gute Ergebnisse (i.e. Komprimierung von bis zu 90% möglich). In XML-Datenbanken werden die XML-Dokumente oft in binärer Form abgelegt, was zu verringerten Zugriffszeiten und einem verminderten Datenaufkommen führt. Alternativ zu einer komprimierten Übertragung in IHE-XDS kann das EHR-Dokument optimiert werden. HL7 arbeitet an einer optimierten Form der CDA, die das Datenaufkommen verringern soll, der sogenannten *Green CDA* [164]. Neben dem optimierten CDA-Dokument wird ein CDA-Schema und eine Transformation von der *Green CDA* und vice versa zur Verfügung gestellt. Die *Green CDA* kann so kleiner gemacht werden, da redundante Information nicht immer übertragen werden muss, sondern in der Transformationsinformation schon enthalten ist.

Beim *Information Retrieval* werden in der Regel die relevanten Dokumente indiziert, um ein mehrfaches Durchsuchen der Dokumente zu vermeiden und die Geschwindigkeit der Suche zu erhöhen. Ein Index über die Dokumentinhalte in IHE-XDS erlaubt eine inhaltsbasierte Suche, führt aber zu Datenschutzbedenken, da medizinische Inhalte an einer zentralen Stelle gespeichert werden. Das verteilte Speichern der Daten in IHE-XDS bringt in diesem Fall keine erhöhte Sicherheit. Weiters ist die Erstellung eines Index von XML-Dokumenten mit Schwierigkeiten verbunden. Im Gegensatz zur Suche auf Dokumentenebene werden beim XML-Retrieval nicht das ganze Dokument, sondern nur die relevanten Teile zurückgeliefert. Werden alle möglichen Teile eines XML-Dokuments im Index gespeichert, so muss für jeden Knoten im XML-Dokument ein Index erstellt werden. In [165] wird die Problematik genau beschrieben und auch ein Lösungsansatz zum Minimieren der indizierten Teile wird vorgeschlagen. Wegen der Datenschutzbedenken wurde der Ansatz mit zentralem Index nicht weiter verfolgt.

Alternativ zur Suche mittels XQuery kann direkt in den Java-Instanzen der XML-Dokumente gesucht werden, die durch Parsen der XML-Dokumente erzeugt werden können. Im Rahmen des EHR-Arche-Projekts wurden für einen Patienten 53 Testdokumente angelegt. Da es sich dabei um eine überschaubare Anzahl von Dokumenten handelt, konnten diese nach dem Parsen direkt im Speicher gehalten werden. Statt mit XQuery und XML-Dokumenten, wurde in diesem Fall direkt in den Java-Instanzen mit Java gesucht. In einem Prototypen wurden beim ersten Herunterladen der Dokumente diese mit Java geparkt. Basierend auf den Informationsbedürfnissen wurde ein Index aufgebaut, in dem für jedes Informationsbedürfnis die relevanten Teile des XML-Dokuments hinterlegt sind. Der Index wurde zum Zeitpunkt des

Herunterladens der Dokumente erstellt, es handelte sich um einen temporären lokalen Index. Durch Auswählen eines Informationsbedürfnisses wurden die relevanten Teile aus dem Index dargestellt. Es wurden jedoch nur Suchen nach einzelnen Informationsbedürfnissen implementiert. Verknüpfungen zwischen Informationsbedürfnissen wurden nicht berücksichtigt, auch sind Wert einschränkungen nicht umgesetzt worden. Nachteile dieses Ansatzes sind der große Speicherbedarf des Index und die bei der ersten Suche benötigte Zeit, um den Index aufzubauen. Die Suche in Java-Instanzen lieferte dafür die Resultate einer Suche nach einzelnen Informationsbedürfnissen in einem Bruchteil einer Sekunde zurück. Der Vorteil eines lokalen temporären Index steigt mit der Anzahl der Suchen, die auf dieselben Dokumente abgesetzt wird. Die optimierte metadatenbasierte Suche wurde in diesem Versuch nicht angewendet, es wurden für Testzwecke einfach alle Dokumente eines Patienten heruntergeladen.

Durch die Verwendung einer XML-Datenbank zur temporären Speicherung der zurückgelieferten Dokumente können ebenfalls kürzere Rückgabezeiten erzielt werden. Im Unterschied zu den verwendeten Java-Instanzen und dem auf die Informationsbedürfnisse angepassten Index können bei existierenden XML-Datenbanken die Vorteile von vorhandenen Indizes sowie Optimierungen der XQuery genutzt werden. XML-Datenbanken indizieren die Dokumente beim Speichern in der Datenbank. Bei Tests mit der XML-Datenbank *BaseX* [166] wurde das Ergebnis einer einfachen XQuery-Abfrage (zwei Informationsbedürfnisse) über 53 Testdokumente in 0,4 Sekunden zurückgeliefert, was im Vergleich zu der in der vorliegenden Arbeit präsentierten Umsetzung deutlich kürzer ist. *BaseX* unterstützt auch eine Volltextsuche über XQuery 3.0. Die Erstellung des Index der 53 Testdokumente dauerte kürzer als das Herunterladen der Dokumente vom IHE-XDS. Eine Kombination eines auf die Informationsbedürfnisse abgestimmten Index und der Optimierungen in existierenden XML-Datenbanken kann zu kürzeren Suchzeiten führen.

Durch das Indizieren der zugrundeliegenden Strukturen der EHR-Dokumente müsste nicht immer das ganze Dokument mittels XQuery durchsucht werden, es könnten nur die relevanten Teile aus dem XML-Dokument geladen werden. In [167] wird eine Möglichkeit beschrieben, *Views* von mehreren XML-Dokumenten zu durchsuchen, ohne die *Views* zu instanzieren. In der vorliegenden Arbeit werden XPath und XQuery zum Extrahieren relevanter Teile verwendet. Alternative Lösungen, die schneller auf konkrete Elemente "zugreifen" und diese aus einem XML-Dokument extrahieren, wurden nicht analysiert. Hier liegt jedoch großes Potenzial, um

die Dauer der Suche zu verkürzen.

Semantic-Web-Technologien wurden in der vorliegenden Arbeit nicht für die Suche der EHR-Inhalte in EHR-Dokumenten verwendet. Alternativ zum Aufbau eines Index in Java, der beim Parsen der Dokumente erstellt wird, können die XML-Dokumente beim Parsen in *RDF-Triple* umgewandelt werden (siehe [109]). Dies würde es erlauben, SPARQL-Suchanfragen abzusetzen. Ähnlich wie bei der Verwendung von XML-Datenbanken kann auf existierende, optimierte Tools zurückgegriffen werden, um effizient EHR-Inhalte zurückzuliefern. Beim Umwandeln der XML-Dokumente in *RDF-Triple* könnten direkt Verknüpfungen zu den Archetyp-Knoten generiert werden. Da das *Archetype Repository* schon auf Semantic-Web-Technologien aufbaut, wäre ein Einbinden der Verknüpfungen direkt möglich. Da die EHR-Dokumente in IHE-XDS als XML-Dokumente abgespeichert sind, müssten für eine Suche die EHR-Dokumente jedes Mal aufs Neue in *RDF-Triple* umgewandelt werden. Wie sich diese Umwandlung vor der Suche auf die Laufzeit der Suche auswirkt, wurde nicht untersucht.

Für eine weitere Optimierung der Suche in medizinischen Dokumenten könnte eine Testinfrastruktur ähnlich zu der *INitiative for the Evaluation of XML-Retrieval* (INEX) aufgebaut werden. Durch ein zentrales Zur-Verfügung-Stellen von Testdaten und Suchanfragen könnten unterschiedliche Suchalgorithmen objektiv miteinander verglichen werden. Die vorgestellten alternativen Suchmöglichkeiten wurden teilweise als kleine Prototypen umgesetzt; Vorrangiges Ziel der Arbeit war es aber, das Potenzial der Suche mittels Archetypen zu analysieren und nicht die Suchgeschwindigkeit zu optimieren. Die Testdaten bei INEX umfassen einige Terabyte an Daten, der Aufbau einer solchen Testumgebung für EHR-Dokumente ist aus Datenschutzgründen und wegen Problemen der Anonymisierung der personenbezogenen medizinischen Dokumente nicht einfach. Die 53 ISO/EN-13606-EHR-Extrakte sowie die vorgefertigten Suchanfragen könnten aber als Basis einer Testumgebung herangezogen werden.

#### **6.1.4 Alternative Visualisierungsansätze**

Die Visualisierung der Ergebnisse ist ein wesentlicher Faktor bei der Vermeidung von Informationsüberflutung. Neben der beschriebenen tabellarischen Darstellung sind noch weitere Visualisierungen denkbar. In der vorgestellten tabellarischen Visualisierung wurden

alle Werte in textueller bzw. numerischer Form dargestellt. Für einfache Suchanfragen mit nur einem Parameter könnte der entsprechende Visualisierungstyp aus dem zugrundeliegenden Datentyp ermittelt werden und die Werte damit optimal dargestellt werden. Numerische Werte könnten zum Beispiel als Verlauf dargestellt werden. Für komplexe Suchanfragen mit vielen unterschiedlichen Rückgabewerten ist ein Vorbereiten der Visualisierung erst möglich, wenn alle Dokumente heruntergeladen und geparkt sind. Erst zu diesem Zeitpunkt steht fest, was visualisiert werden muss. Werden zum Beispiel 100 unterschiedliche Parameter gesucht, es existieren aber nur Daten zu zehn davon, ist es sinnvoll, die Visualisierung an diese zehn anzupassen. Auch kann die Visualisierung abhängig von der Anzahl der gefundenen Werte variieren.

Als proof-of-concept wurde eine Android-Applikation umgesetzt (siehe Abbildung 6.1). Numerische Werte werden dabei als Verlauf dargestellt. Im linken Fenster werden Informationsbedürfnisse, die eine numerische Ausprägung haben, von der Wissensbasis geladen und dem Benutzer angezeigt. Es können mehrere Informationsbedürfnisse gleichzeitig ausgewählt werden. Im mittleren Fenster kann der gewünschte Zeitbereich angezeigt werden. Im rechten Fenster werden die Ergebnisse der Suchanfrage, die von der Android-App an den *Document Crawler* gesendet werden, angezeigt. Der *Document Crawler* erkennt, dass die Anfrage von der Android-App gekommen ist, und liefert eine speziell dafür optimierte Rückgabetable zurück, die direkt als grafischer Verlauf dargestellt werden kann.

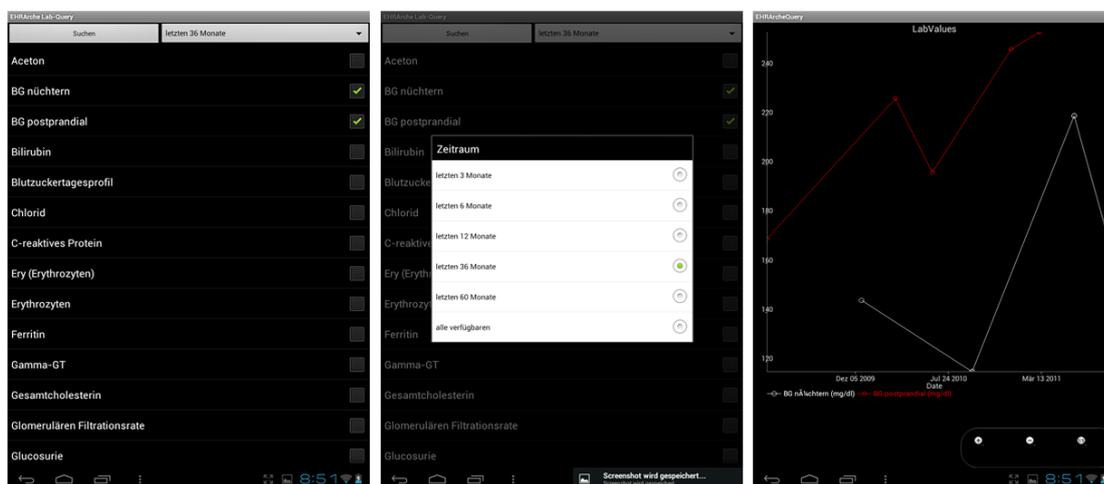


Abbildung 6.1: Android-App zum grafischen Darstellen von numerischen Informationsbedürfnissen.

Die Visualisierung wird im *Document Crawler* für den *Document Consumer* vorbereitet. Dabei werden die Datentypen vereinfacht und nur als String dargestellt, in dieser Umsetzung werden aber auch codierte Werte nicht mehr angezeigt, sondern nur deren *display name* übertragen. Die vereinfachte Ergebnisdarstellung hat den Vorteil, dass der *Document Consumer* das Ergebnis sofort, ohne zusätzliches Wissen über die verwendeten EHR-Standards, darstellen kann. Der Ansatz hat aber auch den Nachteil, dass die Daten im *Document Consumer* nicht mehr weiter verarbeitet werden können. Zusätzliche Informationen aus externen Quellen können nicht nachgeladen werden, da die Codes "gelöscht" wurden, numerische Werte können nicht mehr eindeutig identifiziert werden, da sie als String dargestellt werden und zum Teil die Einheiten im selben String angegeben werden. Bei ähnlichen Problemstellungen sollte im *Document Consumer* wenigstens eine rudimentäre Verarbeitung der unterschiedlichen Datentypen möglich sein. Die Struktur der Ergebnisübermittlung muss dies auch berücksichtigen. Die Datentypen der unterschiedlichen EHR-Standards sind noch nicht harmonisiert. Es muss entweder im *Document Consumer* für alle verwendeten Standards (i.e. CDA, ISO/EN 13606, etc.) eine eigene Verarbeitungslogik implementiert werden oder eine vereinfachte Struktur, die die Anforderungen einer Visualisierung erfüllt und den kleinsten gemeinsamen Nenner der unterschiedlichen EHR-Standards abbildet, erzeugt werden. Die Visualisierung im *Document Consumer* kann dadurch an die Bedürfnisse der Benutzer besser angepasst werden.

In der vorliegenden Arbeit wurde eine generische Darstellung der unterschiedlichen Klassen des Referenzmodells und der Datentypen umgesetzt. Für eine angepasste Visualisierung könnte auf Informationen, die zu den einzelnen Archetyp-Knoten dokumentiert wurden, wie in [131] beschrieben, zurückgegriffen werden. Dies hätte den Vorteil, dass an einer zentralen Stelle Wissen über die Visualisierung zur Verfügung gestellt wird. Wird dies schon zum Zeitpunkt der Erzeugung der Archetypen berücksichtigt, kann eine konsistente Visualisierung gewährleistet werden.

### **6.1.5 Nutzung der inhaltsbasierten Suche im Kontext von ELGA**

Die österreichische ELGA wird auf der Shared-EHR-System-Architektur IHE-XDS aufbauen. Der größte Unterschied der hier vorgestellten Methode im Vergleich zur in ELGA verwendeten Architektur liegt darin, dass in ELGA CDA-Dokumente statt ISO/EN-13606-EHR-Extrakte verwendet werden. Wie in [168] beschrieben können CDA-Dokumente jedoch auch mit Archetypen beschrieben werden.

Für die Umsetzung einer inhaltsbasierten Suche in ELGA müssten alle in ELGA gespeicherten EHR-Inhalte mittels Archetypen oder einer anderen computerverarbeitbaren Form beschrieben werden, was zum jetzigen Zeitpunkt noch nicht der Fall ist. Die vorgestellte Methode kann ebenfalls mit CDA-Dokumenten umgesetzt werden. Die in der vorliegenden Arbeit verwendeten Testdokumente sind im Gegensatz zu den bis jetzt in ELGA spezifizierten EHR-Dokumenten vollstrukturiert. Die medizinischen Inhalte sind bis auf die Ebene einzelner medizinischer Werte definiert. Die CDA-Dokumente von ELGA sind zum Teil nur in CDA Level 2 spezifiziert, d.h. viele medizinische Inhalte werden als Freitext dokumentiert. Da nur die Überschriften der Freitexte mittels Archetypen eingeschränkt werden, könnte mit dem hier präsentierten Ansatz in diesem Fall nur bis auf die Ebene der Überschriften gesucht und auch die Werte nur in dieser Granularität angezeigt werden.

Die im EHR-Arche-Projekt verwendete IHE-XDS-Umgebung verwendet für die Schnittstellendefinitionen ITI-15-, ITI-17- und ITI-18-Transaktionen. In der aktuellen Version von IHE-XDS.b, die auch in ELGA verwendet wird, wurden ITI-15- und ITI-17-Transaktionen durch ITI-41- (*Provide & Register Document*) und ITI-43-Transaktionen (*RRetrieve Document Set*) ersetzt. Die in dieser Arbeit verwendeten Funktionen sind ebenfalls mittels ITI-41- und ITI-43-Transaktionen umsetzbar, der präsentierte Ansatz kann auch in der neuen Version umgesetzt werden.

Die umgesetzte Methode wurde nicht mit mehreren *XDS Affinity Domains* getestet. Da jedoch nur eine herkömmliche metadatenbasierte Suche benötigt wird, welche auch auf *Multi-Affinity-Domain*-Umgebungen anwendbar ist, ist davon auszugehen, dass der beschriebene Ansatz auch mit EHR-Dokumenten, die in unterschiedlichen *Affinity Domains* abgelegt sind, funktioniert.

Die Identifikation der gesuchten EHR-Inhalte wird in ISO/EN-13606-EHR-Extrakten über die Archetyp-Knoten-ID gemacht, die in CDA-Dokumenten nicht vorhanden ist. Die Archetyp-Knoten-ID könnte in den CDA-Dokumenten von vornherein vorgeschrieben werden, was eine Änderung der Implementierungsleitfäden zur Folge hätte. Eine eindeutige Identifikation kann jedoch auch über die Template-ID, die jetzt in den Implementierungsleitfäden verwendet wird, und die Ausprägungen der Code-Elemente berechnet werden. Dieses Wissen muss im *Archetype Repository* zu jedem Knoten gespeichert werden, um eindeutige XPath-Ausdrücke generieren zu können.

Durch die Verwendung von Archetypen können die vorgestellten Methoden neben der Suche

zum Erstellen neuer Archetypen, zur Validierung der EHR-Dokumente und für die automatische Formulargenerierung verwendet werden. Die automatische Formulargenerierung ist zum Beispiel für ein Ärzte- oder Patienten-Portal denkbar, um das Editieren von EHR-Dokumenten und somit Anlegen neuer Versionen zu ermöglichen.

## 6.2 Archetypen und das *Archetype Repository*

Ein wesentliches Problem bei der Nutzung von ISO/EN-13606-Archetypen besteht darin, dass kein Template-Mechanismus wie etwa bei openEHR vorhanden ist. Dies erschwert die Wiederverwendung von Archetypen. So wurden etwa idente Teile der Archetypen in unterschiedliche COMPOSITION-Archetypen kopiert, anstatt diese direkt wiederzuverwenden. Der Spezialisierungsmechanismus, der als Alternative zu den Templates verwendet wurde, ist nicht ausreichend, da jeweils nur ein Archetyp eingeschränkt werden kann. Archetypen, die über *Slots* eingebunden sind, können mit Spezialisierungen nicht eingeschränkt werden.

Der Ansatz, der in der vorliegenden Arbeit etwa im Labor-Archetyp verwendet wurde, um unterschiedliche Ausprägungen über eine Spezialisierung abzubilden, stellte sich als nicht ideal heraus. Die Hierarchie der im EHR-Dokument abgebildeten Inhalte wird dadurch aufgebrochen, was dazu führt, dass bei der Suche nach einem "Laborwert" die Schwesternknoten ebenfalls durchsucht werden müssen, statt nur die Kinderknoten berücksichtigen zu müssen. Die Hierarchie des EHR-Dokuments bildet also nicht mehr die semantische Hierarchie der Daten ab. Durch das Verwenden des *Archetype Repository* kann das Erzeugen der XQuery-Suchanfragen zwar trotzdem noch generisch gelöst werden, die Arbeit mit Archetypen und EHR-Dokumenten wird aber erschwert.

Das *Archetype Repository* beinhaltet strukturierte Information über die Inhalte, die im Shared-EHR-System abgelegt sind. Diese Information kann an unterschiedlichen Stellen herangezogen werden, um die Suche zu verbessern. Es wird etwa verwendet, um die Anzahl der Dokumente, die vom *Document Repository* heruntergeladen werden müssen, zu minimieren, indem die Suchanfrage in Bezug auf die möglichen Dokumenttypen, in denen die gesuchten Inhalte vorkommen können, analysiert wird. In weiterer Folge werden nur Dokumente der relevanten Dokumenttypen heruntergeladen. Das Herunterladen der Dokumente beansprucht einen großen Teil der Zeit, eine Reduktion der heruntergeladenen Dokumente kann demnach zu großen Zeiteinsparungen führen. Weiters wird die Gefahr einer

Informationsüberflutung durch ein Filtern der irrelevanten Dokumente verringert.

Durch ein Abgleichen der Suchanfrage mit dem Wissen im *Archetype Repository* können strukturelle Eigenschaften der EHR-Dokumente bei der Suche berücksichtigt werden. Das Einbinden von Struktur in Suchanfragen bedarf eines genauen Vorwissens über die Struktur der zugrundeliegenden Dokumente. Werden strukturelle Einschränkungen mit den Suchanfragen falsch verknüpft, kann die zusätzliche Information zu schlechteren Ergebnissen führen, da in den falschen Teilen eines strukturierten Dokuments gesucht wird. Die im Hintergrund hergestellte Verknüpfung der Archetypen mit der Suchanfrage kann semantisch falsche Verknüpfungen verhindern.

Das *Archetype Repository* wird mittels Semantic-Web-Technologien umgesetzt. Alternativ könnte auch ein *inverted Index* [169], wie zum Aufbau vieler Suchindizes verwendet, erstellt werden. Zu jedem Suchbegriff werden dabei die jeweiligen Knoten zurückgeliefert. Die Berechnung der *Minimal Information Unit* über gemeinsame Elternknoten wäre in diesem Fall nicht so leicht möglich, da im *inverted Index* keine Baumstruktur wie im *Archetype Repository* vorhanden ist.

Neben ISO/EN-13606-Archetypen kann das *Archetype Repository* auch mit anderen computerverarbeitbaren Wissensartefakten wie etwa HL7-Templates aufgebaut werden. Dazu müsste eine Importroutine, wie sie für ISO/EN-13606-Archetypen geschrieben wurde, implementiert werden. Klassen anderer Referenzmodelle und andere Datentypen als jene des Standards ISO/EN 13606 werden momentan vom *Archetype Repository* nicht unterstützt. Es existiert zum jetzigen Zeitpunkt keine grafische Benutzeroberfläche zum *Archetype Repository*, um Wissen manuell einzutragen, zu ergänzen oder zu exportieren.

Die Schnittstellen zum *Archetype Repository* werden als Java-Webservices implementiert und mittels SOAP-Nachrichten aufgerufen. Im *Archetype Repository* werden keine Informationen über den Status der Anfrage gehalten. Eine Umsetzung des *Archetype Repository* mit RESTful Webservices ist daher möglich.

### **6.3 Die Plug-and-Play Formulargenerierung**

Das *Model*, aus dem die EHR-Dokumente erzeugt werden, wird mithilfe des *comprehensive Archetype* umgesetzt. Es dient dazu, Formulare zu generieren und die jeweiligen

Formulareingaben zu speichern. Aus dem *Model* kann direkt ein EHR-Extrakt generiert werden. Das *Archetype Object Model* (AOM) und der *comprehensive Archetype* haben sich als *Model* jedoch nicht bewährt. Der *comprehensive Archetype* ist sehr speicheraufwendig. So hat etwa ein *Model* eines Arztbriefes, das nur die verpflichtenden Elemente abbildet, über 3 MB. Das Erzeugen des *comprehensive Archetype* dauert ebenfalls mehrere Sekunden. Das Verdoppeln von einzelnen Formularkomponenten ist mit einem großen Overhead verbunden und auch das Abarbeiten der Baumstruktur zum Erzeugen der EHR-Dokumente ist sehr rechenintensiv. Eine einfachere Struktur als *Model* hätte den Implementierungsaufwand des Systems ZK-Arche reduzieren können.

In der neuen Version 1.5 des AOM werden die Archetyp-Knoten-IDs für Archetypen, die über *Slots* eingebunden werden, separat abgelegt. Im *comprehensive Archetype* müssen die Archetyp-Knoten-IDs daher nicht mehr umbenannt und eindeutig gemacht werden.

Ein Hauptmotiv für die Entwicklung des ZK-Arche-Systems lag darin, eine pfadbasierte Implementierung ähnlich zu [54] mit ISO/EN-13606-Archetypen zu testen. Neben dem Erzeugen der EHR-Extrakte aus dem *Model* wurde diese zweite Möglichkeit, EHR-Extrakte aus Formulareingaben zu erzeugen, umgesetzt. Basierend auf der Annahme, dass jede Formularkomponente durch einen eindeutigen Pfad beschrieben werden kann, wurde ein Ansatz implementiert, um aus einer Liste aus Pfaden und den dazu eingegebenen Werten (i.e. Schlüssel-Wert-Paare) ein archetypkonformes EHR-Extrakt zu erzeugen.

Um aus den Eingaben der *View* unabhängig vom *Model* und dem *Controller* EHR-Dokumente zu erzeugen, wurde zu jeder Formularkomponente eine ID abgelegt, über die der zugrundeliegende Archetyp-Knoten unabhängig vom verwendeten *Model* eindeutig referenziert wird. Die eindeutige ID wird im Folgenden als “Schlüssel” und der eingegebene Wert als “Wert” bezeichnet. Der ADL-Pfad kann nicht direkt als Schlüssel verwendet werden, da sich Knoten in einem Formular wiederholen können, was mit dem ADL-Pfad nicht darstellbar ist. Der ADL-Pfad wurde daher in einen W3C-XPfad umgewandelt, der verdoppelte Knoten mittels einer fortlaufenden Nummer in eckigen Klammern unterscheidbar macht. Als Wert wird der vom Benutzer eingegebene Wert angegeben. Ein Beispiel eines Schlüssel-Wert-Paares ist:

**Schlüssel:** /EHR\_EXTRACT/all\_compositions[archetype\_id= 'CEN-EN13606-COMPOSITION.Laboratory\_findings.v1/at0000' and @xsi:type=' COMPOSITION' ][2]/rc\_id[@xsi:type=' II' ]/extension

**Wert:** 1.2.3.4

Der Schlüssel bezieht sich auf die eindeutige ID einer COMPOSITION in einem Laborbefund. Der Zähler 2 zeigt an, dass dieser Knoten im Formular schon einmal verwendet wird und es sich hier um die zweite Ausprägung handelt. Die Verwendung von XPath als Schlüssel hat den Vorteil, dass existierende XML-Tools für das Parsen und Verarbeiten des Schlüssels verwendet werden können. Die Schlüssel könnten auch direkt für die Validierung des erzeugten EHR-Dokuments herangezogen werden.

Die Schlüssel-Wert-Paare werden in einer *IndexedHashMap* gespeichert. Die *IndexedHashMap* ist eine *HashMap*, die es ermöglicht, Nachbareinträge von Einträgen zu finden. Beim Erzeugen der *IndexedHashMap* werden nur jene Werte abgelegt, die vom Benutzer auch wirklich eingegeben werden. Sobald ein Wert eingegeben wird, werden auch für die darüberliegenden Knoten Einträge in die *IndexedHashMap* eingefügt. Ebenso werden beim Löschen von Einträgen die nicht mehr benötigten Einträge der *IndexedHashMap* gelöscht.

Es wurde derselbe Mechanismus wie in [170, 54] beschrieben verwendet, um aus der *IndexedHashMap* ein EHR-Extrakt zu erzeugen. Im Gegensatz zu der dort verwendeten ORACLE(R)-PL/SQL-Prozedur wird die *IndexedHashMap* in Java abgearbeitet und ein XML-Dokument aufgebaut.

Das Arbeiten mit der *IndexedHashMap* erwies sich als sehr umständlich. Beim Löschen oder Verdoppeln von Formularkomponenten mussten die jeweiligen Einträge erst gefunden und dann gelöscht bzw. verdoppelt werden. Beim Eintragen von Werten musste immer der ganze Unterbaum in die *IndexedHashMap* eingetragen werden. Statt die *IndexedHashMap* beim Befüllen der *View* zu erzeugen, könnte sie sehr leicht aus dem *Model* generiert werden, sofern die zugrundeliegenden Archetyp-Pfade bekannt sind. In einer als MVC umgesetzten Architektur, wie sie für das ZK-Arche verwendet wurde, ist eine Erzeugung der EHR-Dokumente aus dem *Model* im *Controller* anzuraten, sofern im *Model* die zugrundeliegenden Archetyp-Knoten referenziert sind.

## **6.4 Die XML-Schema-Validierung**

Für die semantische Validierung von EHR-Dokumenten wurde XML-Schema als Alternative zu Schematron eingesetzt. Dies hat den Vorteil, dass die semantische und die syntaktische

Validierung auf derselben Technologie aufbauen, die hierarchische Struktur der Dokumente klar ersichtlich ist und zahlreiche Tools für XML-Schema existieren. Die Nachteile des präsentierten Ansatzes können wie folgt zusammengefasst werden. Das generische XSLT-Skript deckt nicht alle möglichen Archetypen ab, manche Einschränkungen müssen pro Archetyp hinzugefügt werden, Vererbung kann in XML-Schema nur bedingt verwendet werden, wodurch eine automatische Generierung des XML-Schemas empfohlen wird. Weiters ist es in XML-Schema nicht möglich Zusammenhänge zwischen mehreren Elementen auszudrücken, wie zum Beispiel der systolische Blutdruckwert muss größer als der diastolische Blutdruckwert sein. Im Gegensatz zu Schematron kann bei XML-Schema nicht zwischen Warnungen und Fehlern unterschieden werden, dies wird bei CDA-Implementierungsleitfäden öfter genutzt, da so unterschieden werden kann, ob ein Element vorhanden sein muss oder nur sollte.

Der Prototyp zum Erzeugen der XML-Schemata aus Archetypen validiert in der jetzigen Implementierung nur die Struktur des Archetyps, wobei immer ein ganzes Dokument validiert wird. Die Validierung der Ausprägungen und der regulären Ausdrücke ist nicht implementiert. Werden im Archetyp einzelne Terminologien eingeschränkt, kann nur das Codesystem selber validiert werden, für die Validierung der Codes müsste ein Terminologieserver zur Verfügung stehen. Für die Erzeugung des XML-Schema kann derselbe Mechanismus verwendet werden, der zur Erzeugung der Formulare und der Testdokumente verwendet wird. Statt Formularkomponenten wird in diesem Fall ein XML-Schema generiert.

## Zusammenfassung und Ausblick

Archetypen werden in der vorliegenden Arbeit als zentrale Komponente für das Arbeiten mit EHR-Dokumenten herangezogen. Der Hauptfokus der Arbeit liegt auf der Umsetzung einer inhaltsbasierten Suche in einem Shared-EHR-System mit dezentraler Datenhaltung und zentraler Metadatenkomponente. Neben der Suche werden Archetypen außerdem verwendet, um automatisch Formulare zur Erzeugung archetypkonformer EHR-Dokumente zu generieren und diese EHR-Dokumente zu validieren.

IHE-XDS ist eine Architektur zur Implementierung von *Shared-EHR-Systemen* mit dezentraler Datenhaltung und zentraler Metadatenkomponente, die national und international immer mehr an Bedeutung gewinnt. So wird etwa die in Österreich geplante nationale Gesundheitsakte ELGA mittels IHE-XDS umgesetzt. In IHE-XDS wird momentan nur eine Suche über Metadaten wie etwa Dokumenttyp, Zeitbereich, in dem die Dokumente erstellt wurden, etc. unterstützt. Existierende Ansätze zum Anbieten einer inhaltsbasierten Suche setzen auf zentrale Indizes, wodurch die Vorteile der dezentralen Datenhaltung und die damit verbundene größere Datensicherheit ausgehebelt werden.

In der präsentierten Methode zur inhaltsbasierten Suche wird mithilfe des in Archetypen beschriebenen Wissens ein Index über den Aufbau der abgelegten EHR-Dokumente erzeugt. Dieses Wissen wird durch das *Archetype Repository* allen Akteuren in IHE-XDS zugänglich gemacht. Das *Archetype Repository* wird verwendet, um im ersten Schritt die erlaubten Begriffe für eine thesaurusbasierte Suchanfrage zu generieren. Aus diesen Begriffen wird eine

inhaltsbasierte Suchanfrage generiert, wobei mehrere Begriffe mittels UND und ODER zu komplexen Suchanfragen, wie sie etwa zum Erzeugen einer *Patient Summary* benötigt werden, zusammengesetzt werden können. Die inhaltsbasierte Suchanfrage wird mithilfe des *Archetype Repository* in eine herkömmliche metadatenbasierte Suchanfrage und eine XQuery-Suchanfrage umgewandelt. Die metadatenbasierte Suchanfrage wird an das *IHE-XDS-Document Registry* gesendet, um relevante Dokumente zu finden und vom *Document Repository* herunterzuladen. Die erzeugte XQuery wird in diesen Dokumenten angewendet, um die gewünschten Inhalte zu suchen und die Suchergebnisse zu erzeugen.

Da eine herkömmliche metadatenbasierte Suche verwendet wird, kann die vorgestellte Methode in existierende IHE-XDS-Umgebungen eingebunden werden. Es sind keine Änderungen an der IHE-XDS-Spezifikation notwendig und die Methode kann auch nur von einzelnen Akteuren umgesetzt werden, ohne die anderen beteiligten Akteure zu beeinflussen. Voraussetzung dafür ist jedoch, dass die Dokumente in strukturierter Form abgelegt sind und mittels Archetypen beschrieben sind. Abhängig von der Granularität der Archetypen und somit der beschriebenen EHR-Dokumente variiert die Granularität der inhaltsbasierten Suchanfragen und der zurückgelieferten Suchergebnisse.

Es kann davon ausgegangen werden, dass die Geschwindigkeit der in der vorliegenden Arbeit vorgestellten inhaltsbasierten Suche nicht mit den Umsetzungen basierend auf einem zentralen Index mithalten kann, da die einzelnen EHR-Dokumente immer erst heruntergeladen und dann durchsucht werden müssen. Eine Evaluierung der Suche im Rahmen des EHR-Arche-Projekts zeigt jedoch, dass sie ausreichend schnell und stabil ist.

Um die Zusammenarbeit von GDAs und Technikern bei der Erstellung von Archetypen zu erleichtern, wurde eine Methode zum iterativen Entwickeln von Archetypen vorgestellt. Archetypen werden dabei von Technikern basierend auf der initialen Planung erzeugt und mithilfe einer automatischen Generierung von Formularen aus Archetypen werden den GDAs bekannte Konzepte wie Formulare und Dokumente zur Verfügung gestellt, um die abstrakten Konzepte, die in Archetypen abgebildet sind, zu evaluieren. Gewünschte Änderungen werden gemeinsam von GDAs und Technikern geplant und in den Archetypen abgebildet, womit der Zyklus von neuem beginnt. 133 ISO/EN-13606-Archetypen wurden erzeugt und im *Archetype Repository* freigegeben.

Für das Erzeugen der in der Suche verwendeten Testdokumente sowie der benötigten

Archetypen mithilfe des iterativen Archetypen-Entwicklungszyklus wurde eine Methode zum automatischen Generieren von Formularen aus Archetypen vorgestellt. Die Methode wurde in der Webapplikation ZK-Arche umgesetzt und baut auf dem *Model-View-Controller*(MVC)-Pattern auf, wobei das *Model* mit dem *AOM* realisiert wurde. Es wurden 82 ISO/EN-13606-EHR-Extrakte für zwei Patienten erzeugt. Im ZK-Arche-System wurden die nötigen Schnittstellen implementiert, um als *IHE-XDS-Document Source* zu dienen.

Für das Validieren der EHR-Dokumente wurde eine Methode basierend auf XML-Schema vorgestellt. Das XML-Schema wurde automatisch aus den Archetypen generiert. Das Problem der *unique particle attribution constraint rule* in XML-Schema wird durch eine spezielle Namensgebung umgangen, die mittels generischem XSLT-Skript auf die EHR-Dokumente angewendet wird. Für die ISO/EN 13606 existieren zwei unterschiedliche Implementierungen, die das XML-Schema automatisch aus Archetypen erzeugen, für die CDA wird das XML-Schema manuell erzeugt.

Archetypen erwiesen sich für die in der vorliegenden Arbeit präsentierten Anwendungsszenarien als geeignet. Der große Vorteil von Archetypen gegenüber textuellen Beschreibungen ist die computerverarbeitbare Form. Es existiert ein Open-Source-ADL-Parser, der es ermöglicht, die hierarchische Struktur von Archetypen einfach abzuarbeiten. Dies erlaubt es zum Beispiel, die Wissensbasis direkt aus den Archetypen automatisch aufzubauen oder Formulare zur Dateneingabe direkt aus Archetypen zu generieren. Die ISO/EN 13606 legt den Fokus auf den Datenaustausch, trotzdem wäre ein Mechanismus ähnlich der openEHR-Templates wünschenswert, um das Arbeiten mit ISO/EN-13606-Archetypen zu erleichtern.

Die präsentierte Methode zur inhaltsbasierten Suche bei dezentraler Datenhaltung mit zentraler Metadatenkomponente wurde im Rahmen eines FWF-Projekts entwickelt. Es ist geplant die Methode in einem Folgeprojekt auf die in Österreich geplante ELGA und CDA-Dokumente anzuwenden. Die erzeugten ISO/EN-13606-Archetypen sind zum größten Teil bis auf die elementare Ebene spezifiziert. Für den Einsatz in ELGA muss der Ansatz mit semistrukturierten Daten (i.e. CDA Level 2) umgesetzt werden.

Der vorgestellte Prototyp verdeutlicht das Potenzial der Methode, für eine routinemäßige Umsetzung müssen jedoch noch einige Punkte überarbeitet werden. Das Erzeugen der

vorgefertigten Suchanfragen muss durch ein visuelles Suchinterface unterstützt werden, die Zeit zum Extrahieren der gewünschten Teile eines EHR-Dokuments muss minimiert sowie die Darstellung der Ergebnisse effizienter gestaltet werden.

# Abbildungsverzeichnis

1.1	Überblick über ein Shared-EHR-System. . . . .	2
1.2	Archetypen als zentrale Komponente unterschiedlicher Anwendungen in Shared-EHR-Systemen. . . . .	5
2.1	Die vier Typen von EHRs laut ISO/TR 20514 (aus [12]). . . . .	10
2.2	Wissen und Information im Ein-Modell-Ansatz (oben) und Zwei-Modell-Ansatz (unten). . . . .	13
2.3	UML-Modell des ISO/EN-13606- <i>Extrakt</i> -Pakets (aus [26]). . . . .	16
2.4	Dezentrale Datenhaltung mit zentraler Metadatenkomponente. . . . .	35
2.5	Überblick über die zentralen Aktivitäten von IHE (Grafik von www.ihe.net). . . . .	36
2.6	Überblick der Akteure und Transaktionen in IHE-XDS (aus [16]). . . . .	38
2.7	Suchfenster für eine Standard-IHE-XDS-Suchanfrage in einem <i>Document Consumer</i> . . . . .	40
2.8	Suchergebnis einer Standard-IHE-XDS-Suche im <i>Document Consumer</i> . . . . .	41
2.9	Beispiel zur booleschen Suchanfrage. . . . .	43
2.10	Umgang mit Zeitintervallen im QED-Profil (aus [117]). . . . .	53
3.1	Übersicht der dezentralen Datenhaltung mit zentraler Metadatenkomponente mit zusätzlicher inhaltsbasierter Suche. . . . .	59
3.2	Zusammenhang inhaltlicher Suchbegriff, Wissensbasis, Archetyp und Dokumenttyp. . . . .	65
3.3	Auflösen der UND- und ODER-Verknüpfung anhand einer exemplarischen Baumstruktur. . . . .	67
3.4	Tabellarische Visualisierung der Konjunktionen der disjunkten Normalform. . . . .	69
3.5	Überblick des iterativen Entwicklungszyklus für Archetypen. . . . .	70
3.6	Zusammenhang zwischen <i>Model</i> , <i>View</i> und <i>Controller</i> . . . . .	74
3.7	Semantische Validierung mittels XML-Schema, adaptiert von [141]. . . . .	78

3.8	Beispiel zur <i>Unique Particle Attribution Constraint Rule</i> von XML-Schema, adaptiert von [141]. . . . .	79
4.1	Der Arztbrief-Archetyp mit allen eingebundenen Archetypen. . . . .	89
4.2	<i>Slots</i> und Spezialisierungen im Archetyp "Familienanamnese". . . . .	90
4.3	HTML-Darstellung der EHR-Dokumente in ZK-Arche. . . . .	92
4.4	Beispiel eines Arztbrief-Formulars in ZK-Arche. . . . .	93
4.5	Akteure in der IHE-XDS-Umgebung SENSE. . . . .	97
4.6	Umsetzung der Suchoberfläche im <i>Document Consumer</i> . . . . .	99
4.7	Dialogbox der <i>Document Source</i> um Dokumente ins <i>Document Repository</i> hochzuladen. . . . .	100
5.1	Überblick IHE-XDS-Architektur mit zusätzlichen Akteuren zur Unterstützung einer inhaltsbasierten Suche. . . . .	102
5.2	Überblick über Interaktionen zwischen den Akteuren bei der inhaltsbasierten Suche. . . . .	103
5.3	Überblick über die im <i>Archetype Repository</i> verwendeten RDF-Klassen. . . . .	105
5.4	Benutzeroberfläche zum Erzeugen der inhaltsbasierten Suchanfragen im erweiterten <i>Document Consumer</i> . . . . .	110
5.5	Suchanfragen-Editor im erweiterten <i>Document Consumer</i> (aus [158]). . . . .	115
5.6	"Diabetesklassifikation" im <i>Archetype Repository</i> mit den damit verbundenen Archetypen. . . . .	117
5.7	Formularkomponente für Dokumentation von LDL-Cholesterin. . . . .	121
5.8	ER-Diagramm der Tabellen zum Erzeugen der Suchergebnisse. . . . .	128
5.9	Aufbau der Rückgabetabelle (aus [21]). . . . .	130
5.10	Aufbau der Rückgabetabelle mit Werteinschränkungen. . . . .	131
6.1	Android-App zum grafischen Darstellen von numerischen Informationsbedürfnissen. . . . .	147

# Tabellenverzeichnis

2.1	Gegenüberstellung der Archetyp-unabhängigen Suchen . . . . .	51
5.1	Dauer (in Sekunden) der unterschiedlichen Verarbeitungsschritte mit Mittelwert und Standardabweichung (SD) der sechs Testläufe. . . . .	134

# Verzeichnis der Listings

2.1	Beispiel ADL des Labortests für Glukosestatus. . . . .	19
2.2	Beispiel CDA Level 2 und Level 3 (gekürzt aus [4]) . . . . .	23
2.3	Beispiel der *RDF-Triple* aus [72] . . . . .	29
2.4	Vereinfachte Darstellung eines ISO/EN-13606-EHR-Extrakts in XML . . . . .	31
2.5	Beispiel einer einfachen <i>FLOWR</i> -XQuery-Abfrage . . . . .	33
5.1	Beispiel <i>RDF-Triple</i> des <i>Archetype Repository</i> in XML . . . . .	106
5.2	Beispiel <i>RDF-Triple</i> der Informationsbedürfnisse in XML . . . . .	107
5.3	Auszug der Suchbegriffe vom <i>Archetype Repository</i> . . . . .	109
5.4	Suchanfrage für systolischen UND diastolischen Blutdruck ODER Hba1c . . . .	113
5.5	XQuery-Suchanfrage . . . . .	125
5.6	Ergebnis einer XQuery-Suchanfrage . . . . .	126

# Eigene Publikationen

G. Duftschmid, C. Rinner, M. Kohler, G. Hübner-Bloder, S. Saboor, E. Ammenwerth. The EHR-ARCHE project: Satisfying clinical information needs in a Shared Electronic Health Record system based on IHE XDS and Archetypes, *International Journal of Medical Informatics*; 2013

C. Rinner, M. Kohler, S. Saboor, G. Hübner-Bloder, E. Ammenwerth, G. Duftschmid. Searching for document contents in an IHE-XDS EHR architecture via Archetype-based indexing of document types, *Proceedings of MEDINFO 2013*; Vol. 192, pp. 1092 - 1092; 2013

G. Duftschmid, J. Chaloupka, C. Rinner. Towards plug-and-play integration of archetypes into legacy electronic health record systems: the ArchiMed experience, *BMC Medical Informatics and Decision Making*; 13:11; 2013

G. Hübner-Bloder, G. Duftschmid, M. Kohler, C. Rinner, S. Saboor, E. Ammenwerth. An EHR Prototype Using Structured ISO/EN 13606 Documents to Respond to Identified Clinical Information Needs of Diabetes Specialists: A Controlled Study on Feasibility and Impact, *Proceedings of AMIA 2012 Annual Symposium*; November 3-7 2012, Chicago, pp. 380-389; 2012

C. Rinner, M. Kohler, G. Hübner-Bloder, S. Saboor, E. Ammenwerth, G. Duftschmid. Archetype based search in an IHE XDS environment, *Proceedings of MIE2012*; Vol. 180, pp. 631-635; 2012

C. Rinner, M. Kohler, G. Duftschmid. Archetypen in der elektronischen Gesundheitsakte - potentielle Anwendungsszenarien in ELGA, *Proceedings of eHealth2012*; pp. 139-145; 2012

C. Rinner, M. Kohler, G. Hübner-Bloder, S. Saboor, E. Ammenwerth, G. Duftschmid. EHR-ARCHE – Vermeidung EHR-bedingter Informationsüberflutung mittels des Zweimodell-Ansatzes, *Proceedings of eHealth2012*; pp. 95-101; 2012

M. Kohler, C. Rinner, G. Hübner-Bloder, S. Saboor, E. Ammenwerth, G. Duftschmid. Automatische Generierung von Formularen aus ISO/EN 13606 Archetypen zur Erzeugung Archetyp-konformer EHR-Extrakte, *Proceedings of eHealth2011* (best student paper award); pp. 197-202; 2011

G. Hübner-Bloder, G. Duftschmid, M. Kohler, C. Rinner, S. Saboor, E. Ammenwerth. Clinical situations and information needs of physicians during treatment of Diabetes mellitus patients: a Triangulation Study, *Proceedings of MIE2011*; pp. 369-374; 2011

C. Rinner, M. Kohler, G. Hübner-Bloder, S. Saboor, E. Ammenwerth, G. Duftschmid. Creating ISO/EN 13606 Archetypes based on Clinical Information Needs, *Proceedings of EFMI Special Topic Conference “e-Health Across Borders Without Boundaries” 2011*; pp. 43-49; 2011

M. Stadter, C. Rinner, M. Kohler, G. Duftschmid. Semantische Validierung von ELGA-konformen Laborbefunden mit XML-Schema, *Proceedings of eHealth2011*; pp.

165-168; 2011

M. Kohler, C. Rinner, G. Hübner-Bloder, S. Saboor, E. Ammenwerth, G. Duftschmid. The archetype-enabled EHR system ZK-ARCHE – Integrating the ISO/EN 13606 standard and IHE XDS profile, *Proceedings of MIE2011*; pp. 799-804; 2011

G. Duftschmid, T. Wrba, C. Rinner. Extraction of standardized archetyped data from Electronic Health Record Systems based on the Entity-Attribute-Value Model, *International Journal of Medical Informatics*; Vol. 79(8); pp. 585 - 597; 2010

R. Dietrich, K. Holzer, J. Chaloupka, C. Rinner, G. Duftschmid. Generischer Export von archetyp-konformen openEHR EHR-Extrakten aus einem Gesundheitsinformationssystem, *Proceedings of eHealth2010 - Health Informatics meets eHealth*; pp. 135-141; 2010

C. Rinner. HL7 an den Wiener Universitätskliniken, HL7-Mitteilungen; Nr. 27; p. 18; 2010

C. Rinner, G. Schnizer, R. Iglar, H. Schmid, G. Duftschmid, T. Wrba. Intramuraler Datenaustausch Mittels ELGA-basierter CDA-Dokumente am Beispiel des AKH Wien, *Proceedings of eHealth2010 - Health Informatics meets eHealth*; pp. 35-41; 2010

C. Rinner, S. Janzek-Hawlat, S. Sibinovic, G. Duftschmid. Semantic validation of standard based electronic health record documents with W3C XML Schema, *Methods of Information in Medicine*; Vol. 49(3), pp. 271-280; 2010

G Hübner-Bloder, G. Duftschmid, M. Kohler, C. Rinner, S. Saboor, E. Ammenwerth. Systematische Erhebung der Informationsbedürfnisse von Ärzten bei der Behandlung von Diabetes mellitus Patienten, *Proceedings of GMDS2010*; pp. 332-334; 2010

C. Rinner, G. Duftschmid. Validation of standardized and structured EHR data in the Dual Model Approach, *Proceedings of MIE 2009*; pp. 118-119; 2009

C. Rinner, G. Duftschmid. Validieren von auf Zweimodell-Ansätzen basierenden, vollstrukturierten EHR-Daten am Beispiel EN/ISO 13606 und HL7 CDA, *Proceedings of eHealth2009 & eHealth Benchmarking 2009 - Medical Informatics meets eHealth*; pp. 215-221; 2009

## Beteiligung des Autors an Arbeitspaketen

Name des Arbeitspakets	Beteiligung
Erhebung der Informationsbedürfnisse	Die Erhebung der Informationsbedürfnisse und die Befragung der GDAs wurden vom Projektpartner UMIT durchgeführt. Der Autor war bei der Sammlung weiterer schriftlicher Quellen sowie bei der Strukturierung der Informationsbedürfnisse beteiligt
ISO/EN-13606-Archetypen	Die ISO/EN-13606-Archetypen wurden vom Autor basierend auf den Informationsbedürfnissen erzeugt und vom Projektpartner UMIT evaluiert. Der dazu verwendete Iterative Archetyp-Entwicklungszyklus wurde vom Autor entwickelt. Existierende Archetypen wurden von einem Informatiker mit der Unterstützung des Autors gesucht und analysiert.
Testdokumente	Die Testdokumente wurden vom Projektpartner UMIT mithilfe des vom Autor und einem Informatiker gemeinsam implementierten Tools zur Plug-and-Play-Formulargenerierung erzeugt. Das Konzept des Tools zur Plug-and-Play-Formulargenerierung stammt vom Autor. Die Methode zur semantischen Validierung von EHR-Dokumenten wurde vom Autor entwickelt und mithilfe von Studenten umgesetzt.
Inhaltsbasierte Suche	Das grundlegende Konzept und die beteiligten Akteure wurden zum Zeitpunkt der Einreichung des Projekts EHR-Arche festgelegt. Der Autor war bei der Detailplanung involviert.
<i>Document Consumer</i>	Der <i>Document Consumer</i> wurde größtenteils vom Projektpartner UMIT umgesetzt. Vom Autor stammt die Visualisierung der Ergebnisse der inhaltsbasierten Suche.
<i>Document Source</i>	Das Tool zur Plug-and-Play-Formulargenerierung wurde als <i>Document Source</i> verwendet. Diese Funktionalität wurde von einem Informatiker umgesetzt.
<i>Archetype Repository</i>	Die Planung und Umsetzung des <i>Archetype Repository</i> mittels RDF-Triple stammen vom Autor. Der Autor wurde bei der Umsetzung von einem weiteren Informatiker unterstützt.
<i>Document Crawler</i>	Die Umsetzung der inhaltsbasierten Suche sowie die Umsetzung mit XQuery und das Aufbereiten der Ergebnisse wurden vom Autor gemacht. Die metadatenbasierte Suche wurde von einem Informatiker umgesetzt.
Evaluierung	Die Evaluierung der inhaltsbasierten Suche wurde vom Projektpartner UMIT durchgeführt.



# Literaturverzeichnis

- [1] V. N. Stroetmann, D. Kalra, P. Lewalle, A. Rector, J. M. Rodrigues, K. A. Stroetmann, G. Surjan, B. Ustun, M. Virtanen, and P. E. Zanstra. Semantic interoperability for better health and safer healthcare. Deployment and research roadmap for Europe. European Community, 2009. <http://www.empirica.com/publikationen/documents/2009/semantic-health-report.pdf>, Mai 2013.
- [2] IBM Österreich GmbH. Machbarkeitsstudie ELGA. Bundesministerium für Gesundheit und Frauen Familie und Jugend, 2007. [http://www.bmg.gv.at/cms/home/attachments/7/2/0/CH1045/CMS1169796766007/machbarkeitsstudie\\_elga.pdf](http://www.bmg.gv.at/cms/home/attachments/7/2/0/CH1045/CMS1169796766007/machbarkeitsstudie_elga.pdf), Mai 2013.
- [3] S. Herbek, H. A. Eisl, M. Hurch, A. Schator, S. Sabutsch, G. Rauchegger, A. Kollmann, T. Philippi, P. Dragon, E. Seitz, and S. Repas. The Electronic Health Record in Austria: a strong network between health care and patients. *European Surgery*, 44(3):155–163, 2012.
- [4] ELGA GmbH. ELGA - Meine elektronische Gesundheitsakte. <http://www.elga.gv.at/>, Mai 2013.
- [5] Bundesministerium für Justiz. Gesundheitstelematikgesetz 2012 - Elektronische Gesundheitsakte-Gesetz – ELGA-G. [http://www.parlament.gv.at/PAKT/VHG/XXIV/I/I\\_01936/fname\\_271567.pdf](http://www.parlament.gv.at/PAKT/VHG/XXIV/I/I_01936/fname_271567.pdf), Mai 2013.
- [6] T. Christensen and A. Grimsmo. Instant availability of patient records, but diminished availability of patient information: a multi-method study of GP’s use of electronic patient records. *BMC Medical Informatics and Decision Making*, 8:12, 2008.
- [7] W. O. Hackl, A. Hoerbst, and E. Ammenwerth. ‘Why the hell do we need electronic health records?’. EHR acceptance among physicians in private practice in Austria: a qualitative study. *Methods of Information in Medicine*, 50(1):53–61, 2011.
- [8] P. Basch. Data Excess and Document Overload: Barriers and Disincentives to an Interconnectes/Interoperable Healthcare System, 2003. [http://library.ahima.org/xpedio/groups/public/documents/external/bok1\\_024236.pdf](http://library.ahima.org/xpedio/groups/public/documents/external/bok1_024236.pdf), Mai 2013.

- [9] A. Kollmann, S. Sabutsch, and E. Seitz. ELGA wird sichtbar: Das ELGA-Zugangsportal und seine Inhalte. In *4. ELGA-Anwenderforum*, 2012.
- [10] Q. Zeng, J. J. Cimino, and K. H. Zou. Providing Concept-oriented Views for Clinical Data Using a Knowledge-based System. *Journal of the American Medical Informatics Association*, 9(3):294–305, 2002.
- [11] APA. Ärztekammer beharrt auf Freiwilligkeit von ELGA, 2012. <http://derstandard.at/1345166901654/Aerztekaemmer-beharrt-auf-Freiwilligkeit-von-ELGA>, Mai 2013.
- [12] ISO/TC 215. ISO/TR 20514:2005 Health informatics – Electronic health record – Definition, scope and context. International Organization for Standardization. 2005.
- [13] D. Wollersheim, A. Sari, and W. Rahayu. Archetype-based electronic health records: a literature review and evaluation of their applicability to health data interoperability and access. *Health Information Management Journal*, 38(2):7–17, 2009.
- [14] D. Kalra. Barriers, approaches and research priorities for semantic interoperability in support of clinical care delivery. Radboud University Nijmegen Medical Centre; Department of Medical Informatics, 2007. [http://www.semantichealth.org/DELIVERABLES/SemanticHealth\\_D4\\_1\\_final.pdf](http://www.semantichealth.org/DELIVERABLES/SemanticHealth_D4_1_final.pdf), Mai 2013.
- [15] D. Hoy, N. R. Hardiker, I. T. McNicoll, P. Westwell, and A. Bryans. Collaborative development of clinical templates as a national resource. *International Journal of Medical Informatics*, 78 Suppl 1:95–100, 2009.
- [16] Integrating the Healthcare Enterprise (IHE). IHE IT Infrastructure (ITI) Technical Framework Revision 9.0, 2012. [http://www.ihe.net/technical\\_framework/index.cfm#IT](http://www.ihe.net/technical_framework/index.cfm#IT), Mai 2013.
- [17] C. Rinner. Electronic health records (EHRs): Data export of health information systems based on the Entity-Attribute-Value model as CEN prEN 13606 compliant EHR extracts by means of Archetypes. Master thesis, Technische Universität Wien, 2007.
- [18] C. Rinner, T. Wrba, and G. Duftschmid. Publishing relational medical data as prEN 13606 Archetype compliant EHR extracts using XML technologies. In *Tagungsband der eHealth 2007 – Medical Informatics meets eHealth*, Seiten 35–39, 2007.
- [19] C. Rinner, G. Duftschmid, and T. Wrba. CEN prEN 13606 konformer Export von medizinischen Daten aus einem Entity-Attribute-Value basierten Informationssystem. In *Telemedizinführer Deutschland 2008 (Sonderkapitel 'Best of Telemed Berlin')*, volume 9, Seiten 34–38, 2008.
- [20] EHR-Arche. Archetype based electronic health record. <http://www.meduniwien.ac.at/msi/arche/>, Mai 2013.

- [21] E. Ammenwerth and G. Duftschmid. EHR-ARCHE Clinical information needs and archetype-based electronic health records. UMIT Hall i. T, MedUni Wien, 2012. <http://iig.umat.at/projekte/arche/Final%20Report%20EHRArche%20April%202012.pdf>, Mai 2013.
- [22] P. Haas. *Medizinische Informationssysteme und Elektronische Krankenakten*. Springer, 2006.
- [23] P. Gibbons, N. Arzt, S. Burke-Beebe, C. Chute, G. Dickinson, T. Flewelling, T. Jepsen, D. Kamens, J. Larson, J. Ritter, M. Rozen, S. Selover, and J. Stanford. Coming to Terms: Scoping Interoperability for Health Care. Health Level Seven (HL7) EHR Interoperability Work Group, 2007. <http://www.hln.com/assets/pdf/Coming-to-Terms-February-2007.pdf>, Mai 2013.
- [24] International Health Terminology Standard Development Organisation (IHTSDO). SNOMED Clinical Terms. <http://www.ihtsdo.org/snomed-ct/>, Mai 2013.
- [25] T. Benson. *Principles of Health Interoperability HL7 and SNOMED*. Health Informatics. Springer, London, 2010.
- [26] ISO/TC 215. ISO 13606 Electronic health record communication. International Organization for Standardization. 2008.
- [27] R. H. Dolin, L. Alschuler, S. Boyer, C. Beebe, F. M. Beilen, P. V. Biron, and A. Shabo. HL7 Clinical Document Architecture, Release 2. *Journal of the American Medical Informatics Association*, 13(1):30–39, 2006.
- [28] openEHR. The openEHR Foundation. <http://www.openehr.org/>, Mai 2013.
- [29] T. Beale. Archetypes and the EHR. *Studies in Health Technology and Informatics*, 96:238–44, 2003.
- [30] Health Level Seven International (HL7). Specification and Use of Reusable Constraint Templates, Release 2. HL7, 2008. <http://www.hl7.org/v3ballot/html/infrastructure/templates/templates.html>, Mai 2013.
- [31] L. Michelsen, S. S. Pedersen, H. B. Tilma, and S. K. Andersen. Comparing different approaches to two-level modelling of electronic health records. *Studies in Health Technology and Informatics*, 116:113–8, 2005.
- [32] EN 13606 Association. The CEN/ISO 13606 Association site. <http://en13606.org/>, Mai 2013.
- [33] M. R. Santos, M. P. Bax, and D. Kalra. Building a logical EHR architecture based on ISO 13606 standard and semantic web technologies. *Studies in Health Technology and Informatics*, 160(1):161–5, 2010.

- [34] Portal Público B-RES. Base de Registro Eletrônico em Saúde do Estado de Minas Gerais (B-RES) (in portugisischer Sprache). <http://sres.saude.mg.gov.br/>, Mai 2013.
- [35] H. Veseli, G. Kopanitsa, and H. Demski. Standardized EHR Interoperability - Preliminary Results of a German Pilot Project using the Archetype Methodology. *Studies in Health Technology and Informatics*, 180:646–50, 2012.
- [36] H. Demski, C. Hildebrand, A. Brass, S. Jedamzik, and R. Engelbrecht. Improvement of cross-sector communication in the integrated health environment. *Studies in Health Technology and Informatics*, 155:95–100, 2010.
- [37] European Commission. European Patients Smart Open Services (epSOS). <http://www.epsos.eu/>, Mai 2013.
- [38] D. Moner, D. Boscá, and J. A. Maldonado. Work Package 3.5 - Semantic Services Appendix G - CEN EN13606 Technical Specifications 2009. [http://www.epsos.eu/uploads/tx\\_epsosfileshare/D3.5.2\\_Appendix\\_G\\_EN13606\\_Implementation\\_01.pdf](http://www.epsos.eu/uploads/tx_epsosfileshare/D3.5.2_Appendix_G_EN13606_Implementation_01.pdf), Mai 2013.
- [39] ISO/TC 215. CEN/TS 14796 Health informatics – Data types. European Committee for Standardization. 2004.
- [40] ISO/TC 215. ISO 18308:2011 Health informatics – Requirements for an electronic health record architecture. International Organization for Standardization. 2011.
- [41] R. Qamar and A. Rector. Semantic issues in integrating data from different models to achieve data interoperability. *Studies in Health Technology and Informatics*, 129(Pt 1):674–8, 2007.
- [42] E. Sundvall, R. Qamar, M. Nyström, M. Forss, H. Petersson, D. Karlsson, H. Åhlfeldt, and A. Rector. Integration of tools for binding archetypes to SNOMED CT. *BMC Medical Informatics and Decision Making*, 8 Suppl 1:S7, 2008.
- [43] E. Sundvall, M. Nyström, H. Petersson, and H. Åhlfeldt. Interactive visualization and navigation of complex terminology systems, exemplified by SNOMED CT. In *Stud Health Technol Inform*, volume 124, Seiten 851–6, 2006.
- [44] S. Yu, D. Berry, and J. Bisbal. Clinical coverage of an archetype repository over SNOMED-CT. *Journal of Biomedical Informatics*, 45(3):408–418, 2012.
- [45] L. Lezcano, S. Sánchez-Alonso, and M.-A. Sicilia. Associating Clinical Archetypes Through UMLS Metathesaurus Term Clusters. *Journal of Medical Systems*, 36(3):1249–1258, 2012.
- [46] O. Bodenreider. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Res*, 32(Database issue):D267–70, 2004.

- [47] C. Rinner, G. Schnizer, R. Iglar, H. Schmid, G. Duftschmid, and T. Wrba. Intramuraler Datenaustausch Mittels ELGA-basierter CDA-Dokumente am Beispiel des AKH Wien. In G. Schreier, D. Hayn, and E. Ammenwerth, editors, *eHealth2010*, Seiten 35–41, 2010.
- [48] openEHR. openEHR Templates. <https://github.com/openEHR/specifications/blob/master/publishing/architecture/am/tom.pdf>, Mai 2013.
- [49] openEHR. Commercial products and systems based on openEHR. <http://www.openehr.org/shared-resources/usage/commercial.html>, Juli 2012 (seit 2013 offline).
- [50] S. Ahn, S. M. Huff, Y. Kim, and D. Kalra. Quality metrics for detailed clinical models. *International Journal of Medical Informatics*, 82(5):408–417, 2013.
- [51] T. Schuler, S. Garde, S. Heard, and T. Beale. Towards Automatically Generating Graphical User Interfaces from openEHR Archetypes. In A. Hasman, R. Haux, J. VanderLei, E. DeClercq, and F. H. R. France, editors, *Ubiquity: Technologies for Better Health in Aging Societies*, volume 124 of *Studies in Health Technology and Informatics*, Seiten 221–226. I O S Press, Amsterdam, 2006.
- [52] G. Freriks, G. de Moor, and D. Kalra. White paper: Archetype paradigm: an ICT revolution is needed. EUROREC, 2007. <http://www.eurorec.org/files/filesPublic/ArchetypeParadigmFeb2007.pdf>, Mai 2013.
- [53] R. Chen, G. O. Klein, E. Sundvall, D. Karlsson, and H. Åhlfeldt. Archetype-based conversion of EHR content models: pilot experience with a regional EHR system. *BMC Medical Informatics and Decision Making*, 9:33, 2009.
- [54] G. Duftschmid, J. Chaloupka, and C. Rinner. Towards plug-and-play integration of Archetypes into legacy EHR systems: Combining the experiences from two studies. *BMC Medical Informatics and Decision Making*, 13(1):11, 2013.
- [55] S. Arikian, T. Shannon, and D. Ingram. Opereffa - A set of open source openEHR implementation components. <http://opereffa.com/>, Mai 2013.
- [56] A. Brass, D. Moner, C. Hildebrand, and M. Robles. Standardized and flexible health data management with an archetype driven EHR system (EHRflex). *Studies in Health Technology and Informatics*, 155:212–8, 2010.
- [57] ISO/IEC JTC 1/SC 34. ISO/IEC International Standard for Schematron. <http://www.schematron.com>, Mai 2013.
- [58] Alschuler Associates LLC. CDA Validator. <http://www.alschulerassociates.com/validator/>, Mai 2013.

- [59] National Institute of Standards and Technology (NIST). CDA Guideline Validation. <http://xreg2.nist.gov/cda-validation/validation.html>, Mai 2013.
- [60] HL7 Usergroup Switzerland. Schematron rules for HL7 CDA templates. <http://hl7ch.svn.sourceforge.net/>, Mai 2013.
- [61] A. Munoz, R. Somolinos, M. Pascual, J. A. Fragua, M. A. Gonzalez, J. L. Monteagudo, and C. H. Salvador. Proof-of-concept design and development of an EN13606-based electronic health care record service. *Journal of the American Medical Informatics Association*, 14(1):118–29, 2007.
- [62] I. Martinez, J. Fernandez, M. Galarraga, L. Serrano, P. de Toledo, J. Escayola, S. Jimenez-Fernandez, S. Led, M. Martinez-Esproncada, and J. Garcia. Implementation experience of a patient monitoring solution based on end-to-end standards. *Conf Proc IEEE Eng Med Biol Soc*, 2007:6426–9, 2007.
- [63] Z. Tun, L. J. Bird, and A. Goodchild. Validating Electronic Health Records Using Archetypes and XML. CRC for Enterprise Distributed Systems: University of Queensland, 2002. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.24.6743&rep=rep1&type=pdf>, Mai 2013.
- [64] The openEHR foundation. Data Validation. <http://www.openehr.org/wiki/display/dev/Data+Validation>, Mai 2013.
- [65] Ocean Informatics. The openEHR Clinical Knowledge Manager (CKM). <http://openehr.org/knowledge/>, Mai 2013.
- [66] openEHR. ADL Workbench. <http://www.openehr.org/downloads/ADLworkbench/home>, Mai 2013.
- [67] J. T. Fernandez-Breis, M. Menarguez-Tortosa, C. Martinez-Costa, E. Fernandez-Breis, J. Herrero-Sempere, D. Moner, J. Sanchez, R. Valencia-Garcia, and M. Robles. A Semantic Web-based System for Managing Clinical Archetypes. *Conf Proc IEEE Eng Med Biol Soc*, 2008:1482–5, 2008.
- [68] C. Martinez-Costa, J. A. Minarro-Gimenez, M. Menarguez-Tortosa, R. Valencia-Garcia, and J. T. Fernandez-Breis. Flexible semantic querying of clinical archetypes. In *Proceedings of the 14th international conference on Knowledge-based and intelligent information and engineering systems: Part I*, Seiten 597–606. Springer-Verlag, 2010.
- [69] P. Haas, R. Mützner, M. Aschhoff, and B. Rimatzki. Terminologieserver. <http://www.fh-dortmund.de/de/studi/fb/4/personen/lehr/haas/fue/Terminologieserver.php>, Mai 2013.
- [70] V. Bicer, O. Kilic, A. Dogac, and G. B. Laleci. *Archetype-Based Semantic Interoperability of Web Service Messages in the Health Care Domain*, chapter Hershey, PA, USA, Seiten 279–302. IGI Global, 2007.

- [71] O. Kilic, V. Bicer, and A. Dogac. Mapping Archetypes to OWL. Middle East Technical University, 2005. <http://uluru.chime.ucl.ac.uk/files/publications/archetypes/MappingArchetypestoOWLTechnical.pdf>, Mai 2013.
- [72] C. Martinez-Costa, M. Menarguez-Tortosa, J. T. Fernandez-Breis, and J. A. Maldonado. A model-driven approach for representing clinical archetypes for Semantic Web environments. *Journal of Biomedical Informatics*, 42(1):150–64, 2009.
- [73] L. Lezcano, M.-A. Sicilia, and P. Serrano-Balazote. Combining OpenEHR Archetype Definitions with SWRL Rules – A Translation Approach Emerging Technologies and Information Systems for the Knowledge Society. In M. Lytras, J. Carroll, E. Damiani, and R. Tennyson, editors, *Proceedings of the 1st world summit on The Knowledge Society: Emerging Technologies and Information Systems for the Knowledge Society*, volume 5288 of *Lecture Notes in Computer Science*, Seiten 79–87. Springer Berlin / Heidelberg, 2008.
- [74] D. Hunter, J. Rafter, J. Fawcett, E. v. d. Vlist, D. Ayers, J. Duckett, A. Watt, and L. McKinnon. *Beginning XML, 4th Edition*. Wrox Press Ltd., 2007.
- [75] World Wide Web Consortium (W3C). XML Schema 1.1. W3C, 2004. <http://www.w3.org/XML/Schema>, Mai 2013.
- [76] M. Lalmas. *XML Retrieval*, volume 7 of *Synthesis Lectures on Information Concepts, Retrieval, and Services*. Morgan & Claypool, Chapel Hill, North Carolina, 2009.
- [77] World Wide Web Consortium (W3C). XML Path Language (XPath) 2.0 (Second Edition). W3C, 2012. <http://www.w3.org/TR/xpath20/>, Mai 2013.
- [78] World Wide Web Consortium (W3C). XSL Transformations (XSLT) Version 2.0. W3C, 2007. <http://www.w3.org/TR/xslt20/>, Mai 2013.
- [79] World Wide Web Consortium (W3C). XQuery 1.0: An XML Query Language (Second Edition). W3C, 2010. <http://www.w3.org/TR/xquery/>, Mai 2013.
- [80] SAXON. The XSLT and XQuery Processor. <http://saxon.sourceforge.net/>, Mai 2013.
- [81] P. Haas. *Gesundheitstelematik: Grundlagen, Anwendungen, Potenziale*. Springer, 2006.
- [82] Integrating the Healthcare Enterprise (IHE). IHE Profile. <http://www.ihe.net/profiles/>, Mai 2013.
- [83] K. W. Boone. Where in the World is XDS. <http://motorcycleguy.blogspot.co.at/2010/01/where-in-world-is-xds.html>, Mai 2013.
- [84] karoshealth. Innovative EHR Solution Grants Healthcare Providers Access to Health Information of 10 Million Patients, 2008. <http://www.karoshealth.com/2008/10/27/innovative-ehr-solution-grants-healthcare->

providers-access-to-health-information-of-10-million-patients/, Mai 2013.

- [85] karoshealth. Standards-Based Cardiac Solution in Switzerland Delivers Proof of EHR Communication Framework Concept, 2008. <http://www.karoshealth.com/2008/10/27/standards-based-cardiac-solution-in-switzerland-delivers-proof-of-ehr-communication-framework-concept/>, Mai 2013.
- [86] C. Saccavini. The IHE XDS infrastructure for the PHR in Veneto and Abruzzo regions. Consorzio Arsenàl.IT, 2011. <http://www.ihe-italy.org/PDF/Saccavini.pdf>, Mai 2013.
- [87] J. Moehrke. French National EHR based on IHE XDS. <http://healthcaresecprivacy.blogspot.co.at/2011/06/french-national-ehr-based-on-ihe-xds.html>, Mai 2013.
- [88] karoshealth. IHE-Based EHR Solution Provides Secure Access to Complete Clinical History in Lower Austria, 2008. <http://www.karoshealth.com/2008/10/27/ihe-based-ehr-solution-provides-secure-access-to-complete-clinical-history-in-lower-austria/>, Mai 2013.
- [89] T. Schabetsberger, F. Wozak, B. Katt, R. Mair, B. Hirsch, and A. Horbst. Implementation of a secure and interoperable generic e-Health infrastructure for shared electronic health records based on IHE integration profiles. *Studies in Health Technology and Informatics*, 160(Pt 2):889–93, 2010.
- [90] C. McCay, D. Kalra, and R. Worden. Results of Investigating the Transformability Between HL7 V3, openEHR and EN/ISO 13606. NHS Data Standards and Products, 2008. [http://www.healthintersections.com.au/wp-content/uploads/2011/12/HL7-openEHR-13606-Transformability\\_v1.0.pdf](http://www.healthintersections.com.au/wp-content/uploads/2011/12/HL7-openEHR-13606-Transformability_v1.0.pdf), Mai 2013.
- [91] M. Kohler, C. Rinner, S. Saboor, G. Hübner-Bloder, E. Ammenwerth, and G. Duftschmid. The archetype-enabled EHR system ZK-ARCHE – Integrating the ISO/EN 13606 standard and IHE XDS profile. In A. Moen, editor, *MIE 2011*, Seiten 799–804, 2011.
- [92] ITH icoserve. sense - smart eHealth solutions. <http://www.ith-icoserve.com/loesungen/sense-smart-ehealth-solutions/uebersicht/>, Mai 2013.
- [93] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. In *Proceedings of the seventh international conference on World Wide Web 7*, Seiten 107–117. Elsevier Science Publishers B. V., 1998.
- [94] S. Amer-Yahia and M. Lalmas. XML search: languages, INEX and scoring. *SIGMOD Record*, 35(4):16–23, 2006.

- [95] M. W. Berry and M. Browne. *Understanding Search Engines: Mathematical Modeling and Text Retrieval (Second Edition)*. SIAM, Philadelphia, 2005.
- [96] A. Göker. *Information Retrieval: Searching in the 21st Century*. John Wiley & Sons, Ltd., Chichester, West Sussex, 2009.
- [97] A. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.
- [98] M. Manoj and J. Elizabeth. Information retrieval on Internet using meta-search engines: A review. *Journal of Scientific and Industrial Research*, 67(10):739–746, 2008.
- [99] J. Kamps, M. Marx, M. d. Rijke, and B. Sigurbjörnsson. Articulating information needs in XML query languages. *ACM Transactions on Information Systems*, 24(4):407–436, 2006.
- [100] A. Trotman, S. Geva, and J. Kamps. Report on the SIGIR 2007 workshop on focused retrieval. *SIGIR Forum*, 41(2):97–103, 2007.
- [101] L. Guo, F. Shao, C. Botev, and J. Shanmugasundaram. XRANK: ranked keyword search over XML documents. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, Seiten 16–27, 872762, 2003. ACM.
- [102] G. Antoniou and F. van Harmelen. *A Semantic Web Primer (Second Edition)*. MIT Press, 2008.
- [103] P. Hitzler, M. Krötzsch, S. Rudolph, and Y. Sure. *Semantic Web Grundlagen*, 2008.
- [104] K. Pfeiffer, G. Göbel, and K. Leitner. Demand for Intelligent Search Tools in Medicine and Health Care. In H. Blanken, T. Grabs, H.-J. Schek, R. Schenkel, and G. Weikum, editors, *Intelligent Search on XML Data*, volume 2818 of *Lecture Notes in Computer Science*, Seiten 5–18. Springer Berlin / Heidelberg, 2003.
- [105] D. Clunie. A Simplified Medical Document format based on static web pages semantic annotation. <http://dclunie.blogspot.co.at/2012/08/a-simplified-medical-document-format.html>, Mai 2013.
- [106] K. Natarajan, D. Stein, S. Jain, and N. Elhadad. An analysis of clinical queries in an electronic health record search utility. *International Journal of Medical Informatics*, 79(7):515–22, 2010.
- [107] S. Schulz, P. Daumke, P. Fischer, and M. Muller. Evaluation of a document search engine in a clinical department system. *AMIA Annual Symposium Proceedings*, Seiten 647–51, 2008.
- [108] F. Thiessard, F. Mougin, G. Diallo, V. Jouhet, S. Cossin, N. Garcelon, B. Campillo, W. Jouini, J. Grosjean, P. Massari, N. Griffon, M. Dupuch, F. Tayalati, E. Dugas, A. Balvet, N. Grabar, S. Pereira, B. Frandji, S. Darmoni, and M. Cuggia. RAVEL: Retrieval And Visualization in EElectronic health records. *Studies in Health Technology and Informatics*, 180:194–8, 2012.

- [109] H. Liu, X. Q. Hou, G. Hu, J. Li, and Y. Q. Ding. Development of an EHR system for sharing - a semantic perspective. *Studies in Health Technology and Informatics*, 150:113–7, 2009.
- [110] S. Liu, B. Zhou, G. Xie, J. Mei, H. Liu, C. Liu, and L. Qi. Beyond regional health information exchange in China: a practical and industrial-strength approach. *AMIA Annual Symposium Proceedings*, 2011:824–33, 2011.
- [111] C. Pruski and F. Wisniewski. Efficient medical information retrieval in encrypted electronic health records. *Studies in Health Technology and Informatics*, 180:225–229, 2012.
- [112] F. Farfan, V. Hristidis, A. Ranganathan, M. Weiner, and Ieee. XOntoRank: Ontology-Aware Search of Electronic Medical Records. In *Icde: 2009 IEEE 25th International Conference on Data Engineering, Vols 1-3*, IEEE International Conference on Data Engineering, Seiten 820–831. IEEE, 2009.
- [113] C. Ma, H. Frankel, T. Beale, and S. Heard. EHR query language (EQL)–a query language for archetype-based health records. *Studies in Health Technology and Informatics*, 129(Pt 1):397–401, 2007.
- [114] Ocean Informatics. Archetype query builder. <http://www.oceaninformatics.com/Solutions/openehr-solutions/ocean-products/Clinical-Modelling/Ocean-Query-Builder.html>, Juli 2012 (seit 2013 offline).
- [115] S. Sachdeva and S. Bhalla. Semantic interoperability in standardized electronic health record databases. *Journal of Data and Information Quality*, 3(1):1–37, 2012.
- [116] E. Sundvall, M. Nyström, M. Eneling, D. Karlsson, and H. Öрман. Bookmarking Service Considerations for an Archetype-Based EHR Using REST. In A. Moen, editor, *MIE2011*, Seite nicht vorhanden, 2011.
- [117] Integrating the Healthcare Enterprise (IHE). IHE Patient Care Coordination (PCC) Technical Framework Supplement - Query for Existing Data (QED), 2008. [http://www.ihe.net/technical\\_framework/index.cfm#IT](http://www.ihe.net/technical_framework/index.cfm#IT), Mai 2013.
- [118] Health Level Seven International (HL7). HL7/ASTM Implementation Guide for CDA Release 2 -Continuity of Care Document (CCD) Release 1. [http://www.hl7.org/implement/standards/product\\_brief.cfm?product\\_id=6](http://www.hl7.org/implement/standards/product_brief.cfm?product_id=6), Mai 2013.
- [119] R. Spronk. Context issues with the IHE QED profile, 2010. [http://www.ringholm.com/column/Context\\_issues\\_with\\_IHE\\_QED.htm](http://www.ringholm.com/column/Context_issues_with_IHE_QED.htm), Mai 2013.
- [120] Integrating the Healthcare Enterprise (IHE). IHE Patient Care Coordination (PCC) Technical Framework Supplement - Care Management (CM), 2008. [http://www.ihe.net/technical\\_framework/index.cfm#IT](http://www.ihe.net/technical_framework/index.cfm#IT), Mai 2013.

- [121] Integrating the Healthcare Enterprise (IHE). IHE IT Infrastructure Technical Framework Supplement - Multi-Patient Queries (MPQ), 2012. [http://www.ihe.net/technical\\_framework/index.cfm#IT](http://www.ihe.net/technical_framework/index.cfm#IT), Mai 2013.
- [122] Integrating the Healthcare Enterprise (IHE). IHE IT Infrastructure Technical Framework Supplement - On-Demand Documents, 2011. [http://www.ihe.net/technical\\_framework/index.cfm#IT](http://www.ihe.net/technical_framework/index.cfm#IT), Mai 2013.
- [123] F. S. Roque, L. Slaughter, and A. Tkatsenko. A comparison of several key information visualization systems for secondary use of electronic health record content. In *Proceedings of the NAACL HLT 2010 Second Louhi Workshop on Text and Data Mining of Health Documents*, Seiten 76–83. Association for Computational Linguistics, 2010.
- [124] A. A. T. Bui, D. R. Aberle, and K. Hooshang. TimeLine: Visualizing Integrated Patient Records. *IEEE Transactions on Information Technology in Biomedicine*, 11(4):462–473, 2007.
- [125] M. Argüello, J. Des, R. Perez, M. J. Fernandez-Prieto, and H. Paniagua. Electronic Health Records (EHRs) Standards and the Semantic Edge: A Case Study of Visualising Clinical Information from EHRs. In *Proceedings of the UKSim 2009: 11th International Conference on Computer Modelling and Simulation*, Seiten 485–490. IEEE Computer Society, 2009.
- [126] M. Argüello, J. Des, M. J. Fernandez-Prieto, R. Perez, and H. Paniagua. Executing Medical Guidelines on the Web: Towards Next Generation Healthcare Applications and Innovations in Intelligent Systems XVI. *Knowledge-Based Systems*, 22(7):545–551, 2009.
- [127] Massachusetts Institute of Technology. Timeline. <http://simile.mit.edu/timeline/>, Mai 2013.
- [128] amCharts. amCharts - JavaScript Chart Library. <http://www.amcharts.com/>, Mai 2013.
- [129] V. Murdock and M. Lalmas. Workshop on aggregated search. *SIGIR Forum*, 42(2):80–83, 2008.
- [130] Z. Szlávik, A. Tombros, and M. Lalmas. Feature- and query-based table of contents generation for XML documents. In *Proceedings of the 29th European conference on IR research*, Seiten 456–467. Springer-Verlag, 2007.
- [131] G. Kopanitsa. Standard based multiclient medical data visualization. *Studies in Health Technology and Informatics*, 180:199–203, 2012.
- [132] S. Garde, E. Hovenga, J. Buck, and P. Knaup. Expressing clinical data sets with openEHR archetypes: A solid basis for ubiquitous computing. *International Journal of Medical Informatics*, Seiten 334–341, 2007.

- [133] V. Hristidis. Overview of Information Discovery Techniques on EHRs. In V. Hristidis, editor, *Information Discovery on Electronic Health Records*, chapter 3, Seiten 41–54. Chapman & Hall/CRC, 2010.
- [134] openEHR. Archetype Editor. <http://www.openehr.org/downloads/archetypeeditor/home>, Mai 2013.
- [135] Medical-Objects. Clinical Applications, Messaging and Integration. <http://www.medical-objects.com.au/>, Mai 2013.
- [136] J. A. Maldonado, D. Moner, D. Boscá, J. T. Fernández-Breis, C. Angulo, and M. Robles. LinkEHR-Ed: A multi-reference model archetype editor based on formal semantics. *International Journal of Medical Informatics*, 78(8):559–70, 2009.
- [137] C. Larman and V. R. Basili. Iterative and incremental development: A brief history. *Computer*, 36(6):47–56, 2003.
- [138] C. Rinner, M. Kohler, G. Hübner-Bloder, S. Saboor, E. Ammenwerth, and G. Duftschmid. Creating ISO/EN 13606 Archetypes based on Clinical Information Needs. In L. Stoicuvivadar, editor, *EFMI Special Topic Conference STC 2011*, Seiten 43–49, 2011.
- [139] M. Kohler, C. Rinner, G. Duftschmid, S. Saboor, G. Hübner-Bloder, and E. Ammenwerth. Automatische Generierung von Formularen aus ISO/EN 13606 Archetypen zur Erzeugung archetyp-konformer EHR-Extrakte. In G. Schreier, D. Hayn, A. Hörbst, and E. Ammenwerth, editors, *e-Health 2011*, Seiten 197–202, 2011.
- [140] T. Reenskaug. Models-views-controllers. Technical note, Xerox PARC. 1979.
- [141] C. Rinner, S. Janzek-Hawlat, S. Sibinovic, and G. Duftschmid. Semantic validation of standard based electronic health record documents with W3C XML Schema. *Methods of Information in Medicine*, 49(3):271–280, 2010.
- [142] R. Spronk. Eclipse Instance Editor. [http://hl7book.net/index.php?title=Eclipse\\_Instance\\_Editor](http://hl7book.net/index.php?title=Eclipse_Instance_Editor), Mai 2013.
- [143] C. Rinner and G. Duftschmid. Validation of standardized and structured EHR data in the Dual Model Approach. In *MIE 2009*, Seiten 118–119, 2009.
- [144] International Organization for Standardization. Information technology – Document Schema Definition Language (DSDL) – Part 2: Regular-grammar-based validation – RELAX NG. <http://www.relaxng.org/>, Mai 2013.
- [145] G. Hübner-Bloder, S. Saboor, C. Rinner, M. Kohler, G. Duftschmid, and E. Ammenwerth. Clinical situations and information needs of physicians during treatment of Diabetes mellitus patients: a Triangulation Study. In A. Moen, editor, *MIE 2011*, Seiten 369–374, 2011.

- [146] U. Flick. *Triangulation (2. Auflage)*. VS Verlag für Sozialwissenschaften, Wiesbaden, 2008.
- [147] M. Meizoso Garcia, J. L. Iglesias Allones, D. Martinez Hernandez, and M. J. Taboada Iglesias. Semantic similarity-based alignment between clinical archetypes and SNOMED CT: an application to observations. *International Journal of Medical Informatics*, 81(8):566–78, 2012.
- [148] NHS. SVN repository,. <https://svn.connectingforhealth.nhs.uk/svn/public/nhscontentmodels/TRUNK/cm/archetypes/adl/openehr/ehr/>, Mai 2013.
- [149] openEHR. SVN repository. <http://www.openehr.org/svn/knowledge/archetypes/>, Juli 2012 (seit 2013 offline).
- [150] C. Martinez-Costa, M. Menarguez-Tortosa, and J. T. Fernandez-Breis. An approach for the semantic interoperability of ISO EN 13606 and OpenEHR archetypes. *Journal of Biomedical Informatics*, 43(5):736–46, 2010.
- [151] Potix. ZK framework. <http://www.zkoss.org/>, Mai 2013.
- [152] M. Stadter, C. Rinner, M. Kohler, and G. Duftschmid. Semantische Validierung von ELGA-konformen Laborbefunden mit XML-Schema. In G. Schreier, D. Hayn, A. Hörbst, and E. Ammenwerth, editors, *eHealth2011*, Seiten 165–168, 2011.
- [153] T. Schabetsberger, R. Mair, F. Wozak, B. Katt, F. Innerhofer-Oberperfler, and M. Mitterer. health@net — A Case Study from Healthcare Security. In R. Breu and M. Hafner, editors, *Security Engineering for Service-Oriented Architectures*, Seiten 189–222. Springer Berlin Heidelberg, 2009.
- [154] R. Chen and G. Klein. The openEHR Java reference implementation project. *Studies in Health Technology and Informatics*, 129(Pt 1):58–62, 2007.
- [155] B. McBride. Jena: A Semantic Web Toolkit. *IEEE Internet Computing*, 6(6):55–59, 2002.
- [156] M. L. Wilson. *Search User Interface Design*. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool, Chapel Hill, North Carolina, 2012.
- [157] D. Schweiger. Entwicklung eines erweiterten IHE-XDS Document Consumers im Rahmen des Projekts: Archetypen-basierte Elektronische Gesundheitsakte. Bachelor-Arbeit, UMIT - University for Health Sciences, Medical Informatics and Technology, 2012.
- [158] M. Schweitzer. XML-basierte Kommunikation im IHE-XDS-Kontext zur archetypen-basierten Suche. Bachelor-Arbeit, UMIT - University for Health Sciences, Medical Informatics and Technology, 2012.

- [159] C. L. A. Clarke. Controlling overlap in content-oriented XML retrieval. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, Seiten 314–321. ACM, 2005.
- [160] P. Walmsley. FunctX XQuery Function Library. <http://www.xqueryfunctions.com/xq/>, Mai 2013.
- [161] World Wide Web Consortium (W3C). XQuery and XPath Full Text 1.0. W3C, 2011. <http://www.w3.org/TR/xpath-full-text-10/>, Mai 2013.
- [162] A. Trotman and B. Sigurbjörnsson. NEXI, Now and Next. In N. Fuhr, M. Lalmas, S. Malik, and Z. Szlávik, editors, *Advances in XML Information Retrieval*, volume 3493 of *Lecture Notes in Computer Science*, chapter 3, Seiten 41–53. Springer Berlin Heidelberg, 2005.
- [163] F. Özcan, N. Seemann, and L. Wang. XQuery Rewrite Optimization in IBM DB2 pureXML. *IEEE Data Engineering Bulletin*, 31(4)::25–32, 2008.
- [164] Health Level Seven International (HL7). GreenCDA Project. [http://wiki.hl7.org/index.php?title=GreenCDA\\_Project](http://wiki.hl7.org/index.php?title=GreenCDA_Project), Mai 2013.
- [165] K. Hatano, H. Kinutani, M. Watanabe, M. Yoshikawa, and S. Uemura. Determining the Unit of Retrieval Results for XML Documents. In N. Fuhr, N. Gövert, G. Kazai, and M. Lalmas, editors, *Proceedings of the First Workshop of the Initiative for the Evaluation of XML Retrieval (INEX)*, Seiten 57–64, 2002.
- [166] BaseX GmbH. BaseX - The XML Database. <http://basex.org>, Mai 2013.
- [167] F. Shao, L. Guo, C. Botev, A. Bhaskar, M. Chettiar, F. Yang, and J. Shanmugasundaram. Efficient keyword search over virtual XML views. *The VLDB Journal*, 18(2):543–570, 2009.
- [168] D. Moner, A. Moreno, J. A. Maldonado, M. Robles, and C. Parra. Using Archetypes for Defining CDA Templates. *Studies in Health Technology and Informatics*, 180:53–7, 2012.
- [169] S. Büttcher, C. L. A. Clarke, and G. V. Cormack. *Information Retrieval: Implementing and Evaluating Search Engines*. MIT Press, 2010.
- [170] R. Dietrich, K. Holzer, J. Chaloupka, C. Rinner, and G. Duftschmid. Generischer Export von archetyp-konformen openEHR EHR-Extrakten aus einem Gesundheitsinformationssystem. In G. Schreier, D. Hayn, and E. Ammenwerth, editors, *eHealth2010 - Health Informatics meets eHealth*, Seiten 135–141, 2010.

# Lebenslauf

---

## Zur Person

Name Christoph Rinner  
Geboren 12. Dezember 1981, Egg, Vorarlberg  
Nationalität Österreich

---

## Ausbildung

seit 10/2009 Dr.-Studium der techn. Wissenschaften, Informatik  
10/2004 – 05/2007 Masterstudium Medizinische Informatik an der TU Wien  
10/2001 – 10/2004 Bachelorstudium Medizinische Informatik an der TU Wien  
07/1998 – 07/1999 Auslandsschuljahr in Südafrika  
09/1996 – 06/2000 BORG Egg, naturwissenschaftlicher Zweig

---

## Berufserfahrung

seit 08/2006 Projektangestellter am Zentrum für Medizinische Statistik, Informatik und Intelligente Systeme, Medizinische Universität Wien  
2011 und 2013 Vortragender an der Privaten Universität für Gesundheitswissenschaften, Medizinische Informatik u. Technik in Hall i. Tirol im Studiengang Biomedizinische Informatik  
seit 2008 Vortragender an der Medizinischen Universität Wien im Diplomstudium Humanmedizin  
05/2008 – 09/2008 Werkvertrag mit AIT Austrian Institute of Technology GmbH  
12/1996 – 02/2007 Ski- und Snowboardlehrer in den Weihnachts- und Semesterferien  
2004 und 2005 Ferialpraktikum SIMMA Electronic GmbH  
07/2003 – 08/2003 Ferialpraktikum Julius Blum GmbH  
07/2000 – 08/2001 Auslandszivildienst in Bolivien