

Masterthesis

Vergleich medizinischer Modellierungssprachen

Guschal, Florian

Geboren am 08. November 1987 in Zwickau, Deutschland

Studiengang Master Informatik

Westfälische Hochschule Zwickau

Fakultät Physikalische Technik / Informatik

Fachgruppe Informatik

Betreuer, Einrichtung : Prof. Dr. rer. nat. Ralf Laue, Westfälische Hochschule Zwickau

M. Sc. Oliver Arnold , Westfälische Hochschule Zwickau

Abgabetermin: 03.09.2014

Vorwort

Danksagung

Hiermit möchte ich mich bei Prof. Ralf Laue und Oliver Arnold bedanken, die mich während der gesamten Zeit der Arbeitserstellung stets Unterstützt und wichtige Hinweise für den Aufbau der Arbeit gegeben haben.

Außerdem möchte ich mich bei meiner Familie bedanken, die mir den Rücken frei gehalten hat um mir diese Arbeit zu ermöglichen.

Autorenreferat/Abstract

Medizinische Leitlinien zu modellieren und zu formalisieren, sodass sie computergestützt verarbeitet und ausgewertet werden können ist eine komplexe aber wichtige Arbeit. Nur eine solche Ausführbarkeit ermöglicht die beste Patientenversorgung ohne diese aufzuhalten. Dazu werden verschiedene Modellierungssprachen eingesetzt, die domänenspezifische Besonderheiten und Einschränkungen aufweisen.

In dieser Arbeit sollen Sprachen die im klinischen Umfeld Verwendung finden evaluiert werden. Dazu sollen in dieser Arbeit klinische Modellierungssprachen für Prozesse und standardisierte Modellierungssprachen wie UML und BPMN formal verglichen werden.

Heutzutage ist die Computergestützte Ausführung von medizinischen Leitlinien eines der interessantesten Themen der Informatik. Medizinische Leitlinien enthalten Prinzipien und Richtlinien, die Praktiker bei der Diagnose, Therapie oder anderen spezifischen klinischen Behandlungspfaden unterstützen.

Im Rahmen dessen wurde eine umfangreiche Literaturrecherche durchgeführt.

Inhalt

Vorwort.....	1
Danksagung.....	1
Autorenreferat/Abstract.....	2
Kapitel 1: Allgemeines und Einleitung	6
1.1 .1 Motivation.....	6
1.1.2 Ziele der Arbeit	6
1.1.3 Angabe zur Vorgehensweise / Aufbau der Arbeit	7
1.2 Grundlagen und Begriffe	8
1.2.1 HL7 – Health Level 7	8
1.2.2 HL7 Reference Information Model - RIM	8
1.2.3 Medizinische Leitlinie	9
1.2.4 Klinischer Pfad	11
1.2.5 Protégé	12
1.2.6 Skeletaler Plan	12
1.2.7 Onotlogie (Informatik).....	13
Kapitel 2: Problemanalyse	13
Recherche zu den Sprachen	14
Kapitel 3 Einordnung in den Wissenschaftlichen und praktischen Kontext.....	16
3.1 Sprachen zur Repräsentation von Leitlinien	16
3.2 Anforderungen an die Modellierungssprachen	16
3.3 Herkunft verschiedener klinischer Modellierungssprachen.....	19
3.4 Vorstellung der Modellierungssprachen – domänenspezifische Sprachen.....	21
3.4.1 Arden	21
3.4.2 Asbru / Asgaard	31
3.4.3 Klinischer Algorithmus.....	47

3.4.4 EON /Dharma	50
3.4.5 GLARE - Guideline Acquisition, Representation and Execution.....	55
3.4.6GEM – Guideline Elements Model	63
3.4.7GLIF - Guideline Interchange Format	67
3.4.8 (New)GUIDE.....	75
3.4.9 Prestige	77
3.4.10 Prodigy.....	80
3.4.11 PROforma	82
3.5 Vorstellung der Domänenneutralen Sprachen	85
3.5.1 BPMN.....	85
3.5.2 EEPK –erweiterte ereignisgesteuerte Prozesskette	88
3.5.3 UML	90
Kapitel 4: Vergleich der Sprachen.....	93
4.1 Methodik	93
4.2 Allgemeine Sprachelemente	94
4.3 Ablaufplanung und Einbettungsmöglichkeiten der Prozesse	95
4.4 Anwendung der Anforderungen aus dem Grundlagenkapitel.....	97
4.4.1 Anforderung 1	97
4.4.2 Anforderung 2	97
4.4.3 Anforderung 3	98
4.4.4 Anforderung 4	98
4.4.5 Anforderung 5	99
Kapitel 5: Ergebnisse.....	100
5.1 Domänenneutrale Sprachen	100
5.2 Domänenspezifische Sprachen	104
5.2.1 Objektorientierte domänenspezifische Sprachen.....	104

5.2.2 Nicht Objektorientierte domänenspezifische Sprachen	105
Kapitel 6: Zusammenfassung und Ausblick	107
Anlagen	109
A1	109
A2	110
A3	112
A4	112
Symbolverzeichnis / Abkürzungsverzeichnis	118
Tabellen.....	119
Glossar	Fehler! Textmarke nicht definiert.
Quellenverzeichnis	Fehler! Textmarke nicht definiert.
Abbildungsquellen	Fehler! Textmarke nicht definiert.

Kapitel 1: Allgemeines und Einleitung

1.1.1 Motivation

Ein funktionierendes Softwaresystem kann nur dann entworfen werden, wenn genau untersucht wurde, wer das Produkt verwenden soll, welche Aufgaben er damit erledigen soll und welche Informationen notwendig sind um die Aufgabe bestmöglich zu bearbeiten. Besonders im Gesundheitswesen ist es in den letzten Jahren notwendig geworden, die medizinischen Aufgaben bestmöglich und kosteneffektiv zu gestalten, da viele gemeinnützige Krankenhäuser privatisiert wurden. Die für die Aufgaben notwendigen Informationen werden in Modellierungssprachen erstellt, um sie institutionsübergreifend und standardisiert einsetzen zu können. Zugleich werden Computersysteme immer leistungsfähiger und ermöglichen neue Möglichkeiten beim Einsatz in der medizinischen Problemdomäne. In dieser Arbeit sollen deswegen die verschiedenen Modellierungsansätze miteinander verglichen werden und deren Vor- und Nachteile aufgedeckt werden.

1.1.2 Ziele der Arbeit

Ziel der Arbeit ist ein systematischer Vergleich von Prozessmodellierungssprachen in der medizinischen Domäne. Genauer untersucht wird dabei:

1. Wie ist der aktuelle Stand bezüglich des Einsatzes der Sprachen im medizinischen Umfeld?
2. Inwieweit werden mit den Sprachen erstellte Modelle auch tatsächlich derzeit verwendet?
3. Wie sieht im konkreten Falle die Werkzeugunterstützung aus?
4. Inwiefern werden verschiedene Behandlungsvarianten unterstützt?
5. Warum sind die domänenspezifischen Sprachen überhaupt notwendig? Ließen sich diese Prozesse auch in domänenneutralen Sprachen ausdrücken?

Zusätzlich dazu werden die domänenneutralen Modellierungssprachen Erweiterte Ereignisgesteuerte Prozesskette (EEPK), die Business Process Model and Notation (BPMN) und die Unified Modeling Language (UML) zu den Betrachtungen hinzugezogen um einen Vergleich mit domänenneutralen Sprachen zu ermöglichen.

1.1.3 Angabe zur Vorgehensweise / Aufbau der Arbeit

Zu Beginn der Arbeit werden genutzte Begriffe definiert und erläutert.

Danach wird die Methodik und die Ergebnisse der Literaturrecherche zu den Modellierungssprachen vorgestellt. Daraus folgend werden Anforderungen an klinische Modellierungssprachen beschrieben.

Die gefundenen Modellierungssprachen werden daraufhin vorgestellt. Die Vorstellung der Sprache umfasst die Historie der Sprache, die dahinterliegende Idee sowie ihr Ziel, die Sprachelemente, welche Werkzeugunterstützung von den Sprachen angeboten wird und wie ihr aktueller Einsatz im Gesundheitswesen aussieht. Damit sollen die in Kapitel 1.2 genannten Ziele 1,2 und 3 der Arbeit dargestellt werden.

Anhand der Recherche werden die noch relevanten Sprachen ermittelt und danach nach den Anforderungen des Grundlagenkapitels verglichen. Zugleich wurden Sprachen ausgeschlossen, zu denen keine aktuellen Publikationen vorliegen oder die Informationslage unzureichend ist.

Nach der Vorstellung der einzelnen Modellierungssprachen werden diese dann verglichen und deren Möglichkeiten sowie Vor und Nachteile dargestellt. An dieser Stelle wird das Ziel 4 und 5 genauer erläutert.

Zum Schluss erfolgt eine Zusammenfassung der gefundenen Informationen und ein Ausblick gegeben.

1.2 Grundlagen und Begriffe

In diesem Abschnitt werden ausgewählte Begriffe eingeführt, die für das spätere Verständnis der weiteren Kapitel notwendig sind.

1.2.1 HL7 - Health Level 7

Health Level 7, im weiteren HL7 genannt, ist eine nicht profitorientierte Organisation, die 1987 gegründet wurde [41]. HL7 bietet ANSI sowie ISO Standards und ein darauf basierendes Framework, welches die Integration, den Austausch und die Interoperabilität von elektronischen Gesundheitsinformationen vereinfachen soll um die klinische Praxis und das Management zu unterstützen. Ebenso wird die Arden Syntax und das Reference Information Model (RIM) von HL7 gepflegt und bereitgestellt [42]. HL7 hat derzeit mehr als 2300+ Mitgliedern (davon 500 corporate Mitglieder) die 90% der Informationssysteme im Gesundheitswesen abdecken sollen.

1.2.2 HL7 Reference Information Model - RIM

Das Reference Information Model ist ein von HL7 entwickeltes objektorientiertes Datenmodell welches als Grundlage für alle Informationsmodelle und Strukturen der Produkte von HL7 ab Version 3 eingesetzt wird um Informationen zu modellieren und untereinander austauschen zu können [86]. Für den Datenaustausch nutzt das RIM die Extensible Markup Language (XML).

Das RIM besteht im Kern aus 6 Konzepten (oder Klassen) die für alle Arten von Informationen inklusive des Gesundheitswesens genutzt werden können. Mithilfe dieser Klassen werden dann für den Datenaustausch notwendige Konstrukte modelliert.

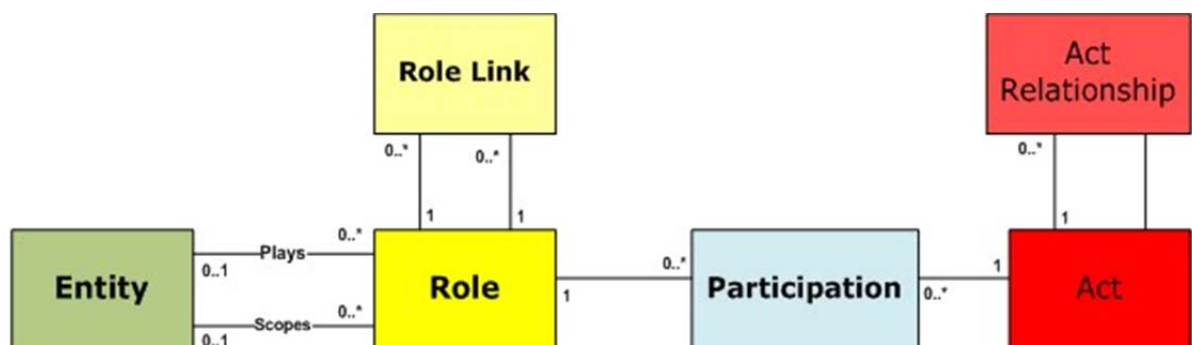


Abbildung 1 : Kernklassen des RIM mit denen alle Informationsaustauschvorgänge ermöglicht werden

1.2.3 Medizinische Leitlinie

Medizinische Leitlinien bilden international die Grundlage [19] für die Entwicklung klinischer Pfade. Im Alltag finden diese Leitlinien in Deutschland eher weniger Berücksichtigung [56]. Diese Leitlinien werden z.B. von der AMWF (Arbeitsgemeinschaft der Wissenschaftlichen Medizinischen Fachgesellschaften), die 1962 gegründet wurde entwickelt und publiziert [47].

Weitere Gesellschaften die Leitlinien in Deutschland publizieren sind:

- AMWF : Arbeitsgemeinschaft der Wissenschaftlichen Medizinischen Fachgesellschaften [4]
- Deutschen Gesellschaft für Kardiologie – Herz- und Kreislaufforschung e. V. [43]
- Ärztliches Zentrum für Qualität in der Medizin [14]
- Kassenärztliche Bundesvereinigung und die Bundesärztekammer [108]
- Nationale Versorgungsleitlinien [68]
- Leitlinien des medizinischen Wissensnetzwerks evidence.de der Universität Witten/Herdecke [64]
- Leitlinien des Berufsverbandes der Augenärzte Deutschlands e. V. (BVA) und der deutschen Ophthalmologischen Gesellschaft e. V. (DOG) [57]

Diese Leitlinien sind Empfehlungen, die auf aktuellen wissenschaftlichen Erkenntnissen sowie in der Praxis bewährten Verfahren beruhen, für die effektivsten Maßnahmen um eine messbare Verbesserung der Gesundheitsversorgung zu bewirken.

Wie werden Leitlinien verwendet?

Leitlinien werden genutzt als [79]:

- Alarmierendes System
- Beratendes System
- Kritisierendes System
- Recherchesystem/Informationsquelle

- Nachträgliche Fallbeurteilung
- Dokumentationssystem
- Lernsystem für neue Mitarbeiter
- Vorgehensplanung (Diagnostik/Therapie)

Sie werden in Entwicklungsstufen S1-S3 eingeteilt [48] . Diese Stufen geben Auskunft über die Qualität der Leitlinie, wobei S3 die höchste Qualitätsstufe darstellt. Die Leitlinien bilden nur eine Empfehlung für den behandelnden Arzt, da keine rechtliche Bindung besteht[48].

Die Kategorien kurz vorgestellt:

- S1 : Leitlinien des Typs S1 werden von einer Expertengruppe im informellen Konsens erarbeitet
- S2k :eine formale Konsensfindung hat stattgefunden
- S2e: Zu dieser Leitlinie wurde eine systematische Evidenzrecherche durchgeführt
- S3: Diese Leitlinie enthält 5 Elemente einer systematischen Entwicklung [89]:
 - o Logik : Ablauflogik der Leitlinie
 - o Evidenzbasierung: heißt hier, dass alle verfügbaren wissenschaftlichen Ergebnisse kritisch diskutiert wurden und je nach Belastbarkeit zur Leitlinienentwicklung herangezogen wurden oder nicht.
 - o Konsensus: bedeutet, dass nach dieser kritischen Sichtung der verfügbaren Studienergebnisse diese in einem gemeinsamen Prozess der Entscheidungsfindung gewichtet und daraus verschiedene Empfehlungsgrade für die jeweiligen Therapieansätze abgeleitet werden.
 - o Entscheidungsanalyse : Entwicklung von Empfehlungsgraden für den Konsensus
 - o Outcome-Analyse : beschreibt das Patienten relevante Gesamtergebnis

S3 Leitlinien werden dabei in regelmäßigen Abständen erneut untersucht um aktuelle medizinische Entwicklungen und Empfehlungen einpflegen zu können.

Leitlinientyp	Anzahl Publikationen	Expertengremium	Evidenzsynthese	Leitlinienentwicklung
S3	128	Alle betroffenen Gruppen	Expertenmeinung	Evidenzbasiert

S2	168	Interdisziplinär besetzte Gruppe	Unsystematische Übersicht	Formaler Konsens
S1	444	Expertengruppe	Expertenmeinung	Informeller Konsens

Tabelle 1 : Übersicht der Leitlinientypen

Obige Tabelle zeigt Zahlen basierend auf den aktuell (Feb 2014) bei der AWMF publizierten Leitlinien [49]. Die Tabelle zeigt, dass nur ca. 30% der Leitlinien in Form einer S3 Leitlinie vorliegen. Bei der Implementierung einer S1 oder S2 Leitlinie besteht ein geringes Risiko noch zu ungenaue oder nicht optimierte Prozesse bereitgestellt zu bekommen.

1.2.4 Klinischer Pfad

Der Begriff klinischer Pfad wird in Deutschland nicht immer gleich interpretiert, da für diesen Begriff zahlreiche Synonyme (Patientenpfad, Klinischer Behandlungspfad, Leitpfad, Indikationspfad, Clinical Pathway, Critical Pathway) existieren, wie der Artikel von Tina Küttner und Norbert Roeder [103] zeigt.

Als kleinster gemeinsamer Nenner können folgende Punkte für diesen Begriff definiert werden:

- Bezug zu einer bestimmten Patientengruppe
- Nutzung durch ein interprofessionelles Behandlungsteam
- Definition eines diagnostischen und therapeutischen Handlungskorridors

Der Begriff „klinischer Pfad“ wird von allen oben genannten Begriffen in der Fachliteratur am allgemeinsten formuliert. Dementsprechend wird im weiteren Verlauf der Arbeit der Begriff „klinischer Pfad“ genutzt. Alle anderen Begriffe sind Spezialformen des klinischen Pfades, oder stellen bestimmte Bedingungen an ihn. Der klinische Behandlungspfad kann zum Beispiel als ein bereits implementierter besonderer klinischen Pfad aufgefasst werden, da dort nach Definition auch Evaluationen zum Zwecke der Verbesserung durchgeführt werden. Der interdisziplinäre Behandlungspfad ist auch eine Spezialform, die als Bedingung einen institutsübergreifende Behandlung vorsieht.

Klinische Pfade werden in dieser Arbeit als institutionsspezifische Implementierung einer medizinischen Leitlinie gesehen, in der die vorhandenen Ressourcen, wie Messinstrumente

und Diagnosegeräte die von einer Leitlinie gefordert werden, an die jeweilige Institution angepasst wurden.

1.2.5 Protegé

Protegé wurde am Institut für medizinische Informatik der Stanford University in Kalifornien konzipiert und ist eine Open Source-Anwendung unter der Mozilla Public License. Mit Protegé werden Ontologien - also Seinszusammenhänge einer Menge von Begrifflichkeiten formal und geordnet konstruiert und dargestellt.

Ursprünglich wurde Protegé dazu entwickelt Wissensdatenbanken beziehungsweise Ontologien zu medizinischen Forschungszwecken zu erstellen, wird aber inzwischen auch in anderen Domänen genutzt. Im informatischen Sinne handelt es sich bei diesen Ontologien um maschinenlesbare Repräsentationen von Wissen aus einem bestimmten Fachbereich (Domäne). In dieser Arbeit wird Protegé von einigen Modellierungssprachen zur Modellierung seiner Wissensstrukturen genutzt.

Die Entwicklungs- und Laufzeitumgebung ist in Java programmiert und damit plattformunabhängig. Die neueste Version von Protegé ist Protegé 5.0 Beta 2.

1.2.6 Skeletaler Plan

Ein skeletaler Plan ist eine generalisierte Vorlage eines klinischen Pfades oder einer medizinischen Leitlinie, die während der Ausführung Freiraum für Anpassungen vom Anwender lässt [82].

Genauer enthält ein Skeletaler Plan eine Sequenz generalisierter Schritte, welche, nach dem sie instanziiert wurden von einem Menschen interpretiert und angewendet werden müssen. Medizinische Leitlinien können z.B. als hochgradig wiederbenutzbare reaktive skeletale Pläne betrachtet werden, deren Details von einem reaktiven Planer über einen langen Zeitraum ausgeführt und verfeinert werden, wenn sie auf einen bestimmten Patienten angewendet werden [81].

Skeletale Pläne werden dazu genutzt um eine Balance zwischen Komplexität und Verständlichkeit von Leitlinien und klinischen Pfaden zu ermöglichen. Bei vielen medizinischen Sachverhalten spielt eine Vielzahl von Faktoren eine Rolle, die für die Behandlung eines Patienten notwendig sind, aber nicht modelliert werden.

Ein Beispiel hierfür ist die Verabreichung eines Medikamentes in einem bestimmten Behandlungsschritt. Der Arzt erhält dabei die Empfehlung ein bestimmtes Medikament zu verabreichen, die Dosierungshöhe wird aber nicht angegeben, da sie zum Beispiel von Alter, Gewicht, Körperbau oder Geschlecht des Patienten abhängig ist und somit vom behandelnden Arzt entschieden werden muss.

Ebenso kann zum Beispiel die zeitorientierte Sprache ASBRU des Asgaard Projektes eine minimale und maximale Behandlungsdauer für einen bestimmten Behandlungsschritt definieren [98].

1.2.7 Ontologie (Informatik)

Eine Ontologie ist in der Informatik ein System von Informationen mit logischen Relationen. Dabei werden sprachlich gefasste und formal geordnete Darstellungen einer Menge von Begrifflichkeiten mit deren Beziehungen untereinander. Sie werden genutzt um „Wissen“ in einer formalen Form in Anwendungsprogrammen und Diensten austauschen zu können. Ontologien werden in der medizinischen Informatik genutzt um medizinisches Wissen strukturiert zu beschreiben.

Kapitel 2: Problemanalyse

Das medizinische Wissen, welches in einer Leitlinie als natürliche Sprache dargestellt wird muss formalisiert werden, damit es computergestützt verarbeitet werden kann. Um das zu erreichen muss die Sprache in Komponenten aufgeteilt werden, die ein Computer interpretieren und ausführen kann. In Folge dessen wurden verschiedene formale Sprachen

und Werkzeuge entwickelt die den speziellen Anforderungen der medizinischen Problemdomäne gerecht werden sollen.

Jede dieser Sprachen konzentriert sich dabei auf bestimmte Aspekte der Modellierung sowie Darstellung der Leitlinie. Viele dieser Sprachen basieren auf einem hierarchischen System aus Komponenten. Komponenten können dabei aus atomaren sowie zusammengesetzten Elementen bestehen. Die zusammengesetzten Komponenten können dabei entfaltet werden um die einzelnen Behandlungsschritte der Komponente zu zeigen. Dies ermöglicht eine abstrahierte Einteilung der Behandlungsschritte, sowie die Möglichkeit einer differenzierten Darstellung der Granularität des Behandlungspfades.

Ansätze hierfür wurden von Wang et al.[28] , Clercq et al. [78] beschrieben. Diese befassen sich mit der minimal notwendigen Zahl von Sprachkonstrukten die zur Darstellung medizinischer Problemstellungen notwendig sind. Von Peleg et.al wurde der am meisten Referenzierte Artikel mit dem Titel „Comparing Computer-interpretable Guideline Models – a case study approach“ [80] als Grundlage für den Vergleich herangezogen.

Recherche zu den Sprachen

Nach initialen Recherchen im Januar und Februar 2014 zu medizinischen Modellierungssprachen wurde auf der Seite von Openclinical.org eine Sammlung vorhandener Sprachen im medizinischen Umfeld gefunden und ausgewertet. Diese Website wurde um die oben genannte Studie „Comparing Computer-interpretable Guideline Models – a case study approach“ [80] herum erstellt und bietet eine Übersicht über implementierungsnahe medizinische Modellierungssprachen. Zugleich wurden über CiteSeer und Google Scholar sowie Pubmed die Datenbanken nach weiterem relevantem Material durchsucht. Dies ergab 11 Modellierungssprachen. Aus diesen Sprachen wurden die Sprachen, die für die Modellierung in der medizinischen Domäne angepasst wurden, herausgenommen. Zusätzlich wurden die 3 Prozessmodellierungssprachen UML, BPMN und EEPK mit in die Betrachtung einbezogen , um einen Vergleich zu aktuellen

Domänenneutralen Sprachen zu ermöglichen, da diese Sprachen in den letzten Jahren viele neue Möglichkeiten bei der Modellierung von domänenspezifischen Fachkonzepten gaben.

Die Ergebnisse der Suchmaschinen wurden so lang untersucht bis keine weiteren relevanten Werke mehr gefunden werden konnten. Die Suche gestaltete sich dabei manchmal nicht sehr einfach, da viele Sprachen an Begriffe der medizinischen Domäne angelehnt wurden und somit sehr viele Ergebnisse gefunden wurden, die kein relevantes Material beinhalteten. So musste bei vielen Publikationen erst der Abstract und die Zusammenfassung gesichtet werden um die Relevanz des Werkes einschätzen zu können. Zugleich lieferten zu spezielle Suchanfragen zu wenig Publikationen oder beinhalteten nicht die am meisten referenzierten Publikationen. Somit konnte nicht ausgeschlossen werden, dass alle Werke gefunden wurden, deswegen wurden allgemeinere Suchanfragen gestellt. Mithilfe der Schlüsselwörter wurden zunächst Werke identifiziert die eine Relevanz (nach Zitationen) für das zu bearbeitende Thema besitzen. Eine Tabelle mit allgemeinen Suchbegriffen und den dazugehörigen am meisten referenzierten Publikationen lässt sich in Anhang A2 finden.

Priorität erhielten dabei die Werke die die meisten Referenzen aufwiesen. Zusätzlich wurden die Publikationen nach dem Erscheinungsjahr ausgewertet um eine Übersicht zu erhalten wie viele Publikationen zu den Werken in den letzten Jahren verfasst wurden. Eine Übersichtstabelle kann dazu in der Anlage A1 gefunden werden.

Die so gefundenen Werke wurden ausgewertet und deren referenzierte Publikationen wurden gesichtet. Anhand dieser Referenzen wurden weitere Werke identifiziert die zum Arbeitsthema passten. Eine Übersicht dazu findet sich in Anlage A3.

Alle so gefundenen Werke wurden dann mithilfe der Zusammenfassung und des Abstracts auf Relevanz überprüft. Weitere Werke der gefundenen Autoren wurden ebenfalls nach relevantem Material durchsucht und zusätzlich wurden häufig auftauchende Autoren kontaktiert um weitere interessante Informationsquellen zu erhalten.

Kapitel 3 Einordnung in den Wissenschaftlichen und praktischen Kontext

3.1 Sprachen zur Repräsentation von Leitlinien

Verschiedene Sprachen zur Repräsentation von Leitlinien werden und wurden für unterschiedliche Zwecke mit unterschiedlichen Zielen für bestimmte Nutzer entwickelt. Um einen Vergleich zu ermöglichen werden dazu nachfolgend Anforderungen an medizinische Modellierungssprachen aufgestellt. Shiffman et al. [88] beschreibt allgemein die Anforderungen an eine Modellierungssprache für eine Leitlinie wie folgt:

- 1. Umfassend : Die Sprache ist dazu in der Lage das gesamte Wissen einer Leitlinie zu beschreiben
- 2. Expressively adequate(Angemessene Ausdrücke): Die Sprache sollte die Komplexität und Nuancen der Medizin vermitteln, während sie informationell gleich zur Leitlinie bleibt.
- 3. Flexibel: Die Sprache sollten mit der Vielfalt und Komplexität einer Leitlinie umgehen können. Die Sprache sollte eine Modellierung von verschiedenen Granularitäten der Leitlinie zulassen um so verschiedene Abstraktionsgrade der Leitlinie zu ermöglichen.
- 4. Verständlich: Die Sprache sollte der natürlichen Sprache der Akteure entsprechen oder ihr nahe kommen. Zugleich sollten Experten ihr Wissen mit wenig Aufwand beschreiben können.
- 5. Austauschbar : Die Sprache sollte über Institutionen hinweg austauschbar sein.
- 6. Wiederbenutzbar : In allen Phasen des Leitlinienlebenszyklus sollte die Sprache wiederbenutzbar sein

3.2 Anforderungen an die Modellierungssprachen

Shiffman [88] beschreibt bereits Anforderungen an eine Modellierungssprache. Diese Anforderungen lassen sich dabei noch weiter verfeinern in Bezug auf Punkt 2 und 3 der von ihm genannten Anforderungen.

Grundsätzlich lassen sich dabei weitere 5 Anforderungen für die Modellierungssprachen definieren.

Die Darstellung erfolgt zielgerichtet im Sinne eines Modells. Gesetzliche Vorgaben und Besonderheiten können hier nicht dargestellt werden. Nach [94] und [62] können die Grundlegenden Anforderungen in der klinischen Anwendungsdomäne in technische und sozio-organisatorische Faktoren eingeteilt werden, die zu großen Teilen diesen Werken entnommen sind. Diese Faktoren bilden die Grundlage für die nachfolgenden Anforderungen. Diese Anforderungen finden sich ebenfalls als Grundlage in dem von Prof. Andreas Gadatsch verfassten Lehrbuch [34] .

Medizinische Leistungserbringungen können als medizinische Geschäftsprozesse aufgefasst werden. Diese Prozesse definieren Grundsätzlich zielgerichtete Abläufe und Tätigkeiten sowie Entscheidungen, die durch Ereignisse und Patientenzustände initialisiert/definiert werden.

Anforderung 1: Eine Sprache zur Modellierung Klinischer Behandlungspfade sollte grundlegende Konzepte der medizinischen Geschäftsprozessmodellierung (Patientenzustand, Behandlungsschritt, Entscheidung und Prozessfluss) enthalten.

Diese Schritte werden meist zeitorientiert und sequenziell ausgeführt, sollten aber ebenso die Möglichkeit besitzen parallele und zeitgleiche Behandlungsschritte ausführen zu können. Spezielle Untersuchungen, wie die Pulsmessung des Patienten unter Last könnten nur so erfolgen.

Anforderung 2: Eine Sprache zur Modellierung Klinischer Behandlungspfade sollte die Modellierung sowohl unbestimmter Reihenfolgebeziehungen als auch zwingend paralleler Beziehungen zwischen Behandlungsschritten ermöglichen.

Medizinische Behandlungspfade besitzen manchmal eine unbestimmte Reihenfolge der Behandlungsschritte. Diese Prozesse sollten die Verfügbaren

Ressourcen(Geräte/Messinstrumente etc.) berücksichtigen, damit diese effizient unter allen Akteuren koordiniert werden können.

Anforderung 3: Eine Sprache zur Modellierung Klinischer Behandlungspfade sollte Konzepte zur Modellierung von Ressourcen (Daten, Dokumente) und Verantwortlichkeiten (Rollen) bereitstellen und eine Verknüpfung mit den Konzepten der Geschäftsprozessmodellierung erlauben.

Klinische Pfade werden auf Grundlage medizinischer Leitlinien und evidenzbasiertem Wissen entwickelt. Geschäftsprozesse werden von einfachen organisatorischen Entscheidungen geprägt. Bei evidenzbasierten Entscheidungen z.B. für die Wahl einer Therapiemethode ist es notwendig mehrere Kriterien einfließen lassen zu können. Diese Entscheidungen werden dann zumeist über die patientenindividuellen Ausprägungen dieser Kriterien erfolgen. Bereits während der Modellierung können aus diesen Kriterien Regeln definiert werden, die einen entscheidungsstützenden Vorschlag generieren. Zugleich sollte es möglich sein eine ärztliche Entscheidung gegen den vom System generierten Vorschlag zu nutzen. Unter entsprechender Dokumentation und Begründung sollten ärztliche Entscheidungen eine höhere Priorität aufweisen.

Anforderung 4: Eine Sprache zur Modellierung Klinischer Behandlungspfade sollte Konzepte zur Darstellung evidenzbasierter Entscheidungen unter Integration entsprechender medizinischer Entscheidungskriterien sowie Regeln zur Vorschlagsgenerierung zur Verfügung stellen.

Klinische Pfade dienen als Instrument zur Kommunikation der beteiligten Akteure. Entsprechend aufbereitet können diese Pfade auch zur Information des Patienten oder dessen angehörigen genutzt werden. Zeitliche Aspekte z.B. die Dauer einer Behandlung oder Therapie besitzen dabei eine besondere Relevanz, da diese für die Organisation und Planung der Ressourcen innerhalb der Klinik wichtig sind. Zusätzlich sollte es auch möglich sein ökonomische Kenngrößen hinzuzufügen um eine umfassende Managementunterstützung zu erlauben.

Anforderung 5: Eine Sprache zur Modellierung Klinischer Behandlungspfade sollte die Integration wirtschaftlicher Aspekte wie zeitliche Terminierung sowie Kenngrößen bezüglich Kosten und Erlöse in das Prozessmodell erlauben.

3.3 Herkunft verschiedener klinischer Modellierungssprachen

Die folgende Grafik zeigt die Entwicklung der Sprachen der medizinischen Domäne in den letzten 40 Jahren. Diese Visualisierung zeigt den Werdegang der verschiedenen Sprachen, sowie aus welchen Sprachen sie entstanden sind. Zugleich zeigt der Zeitstrahl, welche Sprachen heute nicht mehr verwendet werden oder von neueren Sprachen abgelöst wurden. Bei der Untersuchung der Sprachen wurde dabei immer die neuste Sprache oder neuste Sprachversion verwendet. So werden Sprachen, die nicht mehr weiterentwickelt werden durch einen verkürzten Pfeil auf der Zeitlinie dargestellt.

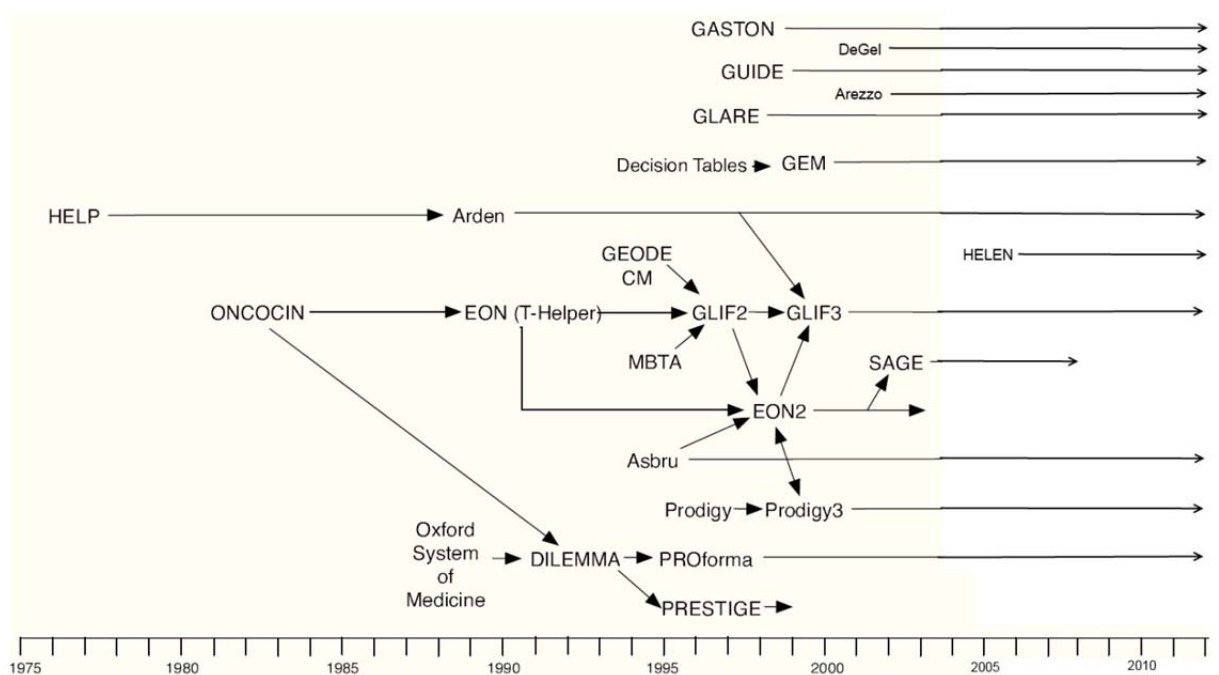


Abbildung 2 : Zeitlinie der Modellierungssprachen , adaptiert nach [21]

Die nachfolgende Tabelle wurde mithilfe der Datenbanksuche von PubMed anhand eindeutiger Schlagwörter (siehe Grundlagenkapitel sowie Anhang A1-A3), sowie www.openclinical.org und dem von Sarshar [93] verfassten Artikel zur Klassifikation von Modellierungssprachen erstellt. Sie zeigt das Jahr der Entstehung sowie das dafür verantwortliche Institut.

Sprache	Entwicklungsort		Jahr der Entwicklung
Arden	HL7 Arden Syntax Special Interest Group and the Clinical Decision Support Technical Committee	USA	1992
Asbru	The Asgaard project, Stanford University University of Vienna	USA	1998
Klinischer Algorithmus	Society for Medical Decision Making	USA	1968
EON /Dharma	Stanford University	USA	1995
EPK	Universität des Saarlandes und Saarbrücken	Deutschland	1992
GASTON	Gaston Medical Group P.A.	Niederlande	2000
GLARE	Università del Piemonte Orientale "Amedeo Avogadro"	Italien	1997
GEM	Yale University School of Medicine	USA	2000
GLIF	InferMed Collaboratory	USA	1996
GUIDE	University of Pavia	Italien	1998
HELEN	Universitätsklinikum Heidelberg	Deutschland	2005
Prestige	Diverse (30 Institutionen aus 8 Ländern)	Europ. Gemeinschaftsprojekt.	1997
Prodigy	University of Newcastle	Großbritannien	1995
PROforma	Imperial Cancer Research Fund	Großbritannien	1998
SAGE	Stanford Medical Informatics	USA	2002
DeGel	Ben-Gurion University of the Negev	Israel	2001

Tabelle 2: Ausgewählte Modellierungssprachen

3.4 Vorstellung der Modellierungssprachen – domänenspezifische Sprachen

3.4.1 Arden

Historie

Die Arden Syntax ist die Weiterentwicklung des HELP „Clinical Decision Support Systems“ (CDSS) des LDS Hospitals in Salt Lake City und des CARE CDSS vom Regenstrief Institute in Indianapolis. HELP bot allgemeine Entscheidungshilfen, basierend auf medizinischen Entscheidungsregeln. Dem klinischen Personal sollten durch integrierte CDS Systeme geholfen werden, indem große Datenmengen automatisiert gefiltert und geprüft werden [52].

1992 wurde die Arden Syntax ein Standard der „American Society and Materials“ (ASTM) und Health Level 7 (HL7) Standard in 1998. Anfang 1999 wurde von HL7 der Arden Syntax 2.0 vorgestellt. Dieser enthielt eine kleine Erweiterung der Sprachbasis um das Stringhandling in Arden zu verbessern [45]. 2005 wurden mit Arden 2.5 Objekte aus der Objektorientierten Programmierung als Datentypen eingeführt.

Arden wird derzeit von einer HL7 „Special Interests Group“ (SIG) betreut und weiterentwickelt. Aktuell wird die Arden 3.0 Syntax entwickelt die keine Rückwärtskompatibilität mehr besitzt, da das Datenformat der von Arden genutzten Medical Logic Modules (MLMs) auf XML umgestellt wird [54].

Idee/Ziel

Die Idee hinter Arden ist die Trennung von Programm und Wissen. Das Wissen wird dabei in einem computerinterpretierbaren Format repräsentiert, soll aber noch vom Benutzer lesbar und verstehbar sein.

Ziel der Arden Syntax ist es medizinische Regeln algorithmisch darzustellen, unter anderem sollen folgende Ziele erfüllt werden:

- Wiederbenutzbarkeit durch Trennung von Daten und Logik
- Übertragbarkeit durch Trennung von Daten und Logik
- Verständlichkeit durch leicht lesbare Syntax

Die Wissensbasis von Arden codiert dazu medizinische Regeln in einzelne unabhängige Module, die Medical Logic Modules (MLMs) genannt werden. Diese Module sind wiederum in Wartungs- (Maintaince), Bibliotheks- (Library), Lokalisierungs- (Resources) und Wissenskategorien (Knowledge) gruppiert.

Mit Arden 3 wird XML, vorher ASCII Text, zur Speicherung der MLMs verwendet. XML wird verwendet um eine bessere Unterstützung für Authoring Tools und die Entwicklung von Interpretern, sowie eine strukturiertere Anzeige der MLMs zu unterstützen. So können die MLMs in XML basierten Nachrichtensystemen übertragen werden, ebenso ist damit eine Übersetzung in andere Repräsentationen wie Programmier- oder Scriptsprachen möglich. Mittels des von HL7 entwickelten „Reference Information Model“ (RIM) soll damit auch das „Curly Braces“ Problem, welches im Abschnitt Datenzugriff genauer erläutert wird, behoben werden. Das RIM ist ein konzeptuelles Informationsmodell für das gesamte Gesundheitswesen zum Austausch von Daten, bestehend aus 4+2 Klassen. Alle Produkte von HL7 ab Version 3 sind davon abgeleitet [45].

Allgemeiner Ablauf des Arden Lebenszyklus

Abbildung 5 zeigt den Lebenszyklus eines in Arden codierten Leitlinienausschnitts.

1. Auswahl der auszuführenden MLMs aus der Wissensbasis. Die Bedingung für die Ausführung eines MLMs ist im evoke Slot eines Modules definiert.
2. Aufruf jedes ausgewählten MLMs. Der Aufruf kann zeitlich verzögert oder periodisch wiederholt geschehen.
3. Einlesen der Daten im data Slot. Die eigentliche Syntax des Datenbankzugriffs steht innerhalb der geschweiften Klammern und ist systemabhängig.
4. Ausführung der Entscheidungslogik. In diesem Schritt können die zuvor gelesenen Daten verarbeitet werden.
5. Schließt die Logik mit positivem Ergebnis ab, wird der Action Slot ausgeführt.
6. Damit ist die Ausführung des MLMs abgeschlossen

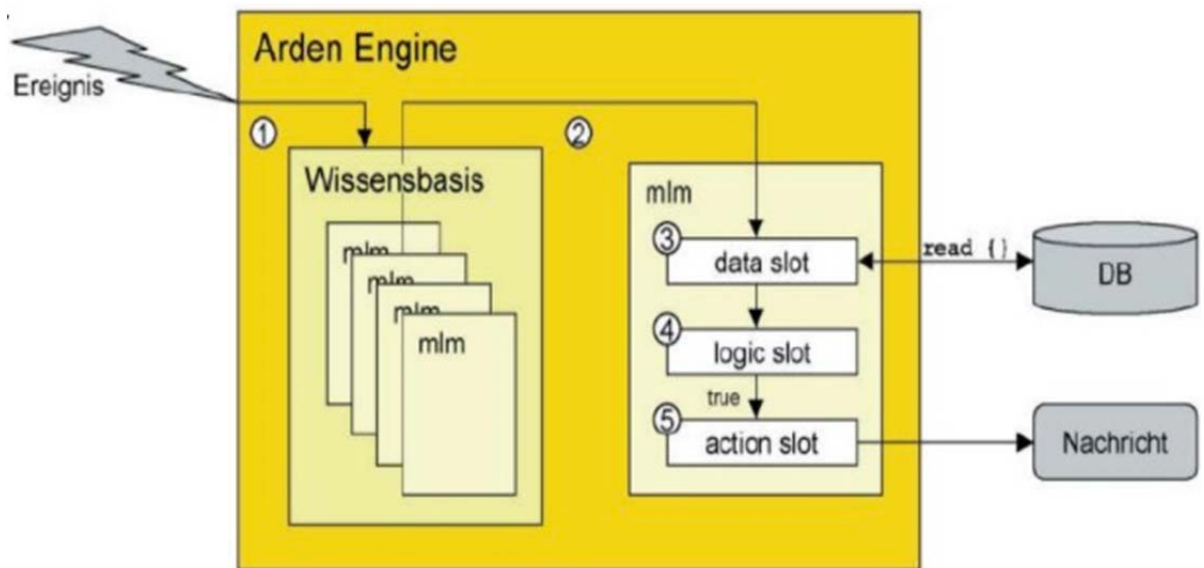


Abbildung 3 : Arden Lebenszyklus (von)

Sprachelemente

Medizinisches Wissen wird durch Medical Logic Modules (MLMs) festgehalten, üblicherweise als ASCII-Datei (Arden2+) oder in XML, wobei die Modellierungselemente die gleichen bleiben. Die Ausführung eines MLM kann aufgrund bestimmter Daten-, Zeit-, und Steuerungsereignisse erfolgen, z.B. beim direkten Aufruf eines MLM über eine Webschnittstelle oder durch ein festgelegtes Ereignis (Event/Trigger) [25].

Jedes einzelne MLM muss dabei genug Informationen besitzen um eine Entscheidung basierend auf den enthaltenen Regeln treffen zu können, z.B. durch zugriff auf eine Datenbank oder direkte Benutzereingaben.

MLMs können wiederum zu Medical Knowledge Packages (MKP) zusammengefasst werden um computergestützte Repräsentationen von abgeschlossenen Teilgebieten darstellen zu können. Dies soll die Wiederverwendbarkeit und Transparenz durch Erläuterungen und zusätzliche Informationsquellen und Internetlinks innerhalb der MLMs erhöhen.

Ein MLM enthält wie erwähnt nicht nur medizinisches Wissen sondern auch Metainformationen über administrative und bibliographische Angaben. Zu diesem Zweck

wurden die MLMs in 4 Kategorien unterteilt in denen diese Informationen gespeichert werden können.

Die 4 Kategorien eines MLMs sind:

Maintaince : Hier finden sich administrative Details zum MLM, wie der Name des MLMs, den Zeitpunkt der letzten Änderung, die Institutionen die das MLM benutzt und die Version des Arden Syntax.

Library: Enthält beschreibende Informationen über den Inhalt des MLMs. Darin enthalten sind textliche Erläuterungen der medizinischen Regeln, Quellen und Links zu Hintergrundinformationen sowie beabsichtigte entscheidungsunterstützende Ziele des MLMs.

Knowledge: Enthält die eigentliche (wenn-dann) Logik des MLMs, also die auszuführenden Aktionen bei erfolgreicher Anwendung der Regel.

Resource: Dient der Lokalisierung der MLMs z.B. für länderspezifische Sprachausgaben der MLMs

Beispiel:

```
maintenance:
    title: Alert on low hematocrit;;
    filename: low_hematocrit;;
    version: 1.00;;
    institution: CPMC;;
    author: George Hripcsak, M.D. (hripcsa@cucis.columbia.edu);;
    specialist: ;;
    date: 1993-10-31;;
    validation: testing;;

library:
    purpose: Warn provider of new or worsening anemia.;;
    explanation: Whenever a blood count result is obtained, the hematocrit is checked
        to see whether it is below 30 or at least 5 points below the previous
        value.;;
    keywords: anemia; hematocrit;;

knowledge:
    type: data-driven;;
    data:
        blood_count_storage := event {'complete blood count'};
        hematocrit := read last {'hematocrit'};
        previous_hct := read last ({'hematocrit'}
            where it occurred before the time of hematocrit);;
    evoke: blood_count_storage;;
    logic:
        /* check that the hematocrit is a valid number */
        if hematocrit is not number then
            conclude false;
        endif;
        if hematocrit <= previous_hct-5 or hematocrit<30 then
            conclude true;
        endif;;
    action:
        write "The patient's hematocrit ("|| hematocrit ||") is low or falling
```

Abbildung 4 : Arden Syntax Beispiel von [77]. Arden Syntax Schlüsselwörter sind dick geschrieben

Slots

Die einzelnen Kategorien wiederum sind in so genannte „Slots“ unterteilt, die jeweils zur Aufnahme weiterer detaillierterer Informationen bestimmt sind.

In der Knowledge Kategorie existieren 5 verschiedene Typen von Slots genannt Data-, Logic-, Language-, Evoke- und Action Slot.

Der Data Slot und Statements :

Der Data Slot wird genutzt um Informationen wie Patientendaten aus den klinischen Informationssystemen abzufragen und um Zuweisungen für die im Knowledge Slot verwendeten Variablen zu unternehmen [77].

Die Variablen werden auf Datenbankelemente der lokalen Institution abgebildet. Dieser Slot bildet die Schnittstelle zwischen Institution und MLM und ist normalerweise an die lokale Organisation und somit deren Datenbankstruktur angepasst.

Evoke Slot:

Der Evoke Slot beschreibt den Kontext in dem ein MLM ausgeführt werden soll [77]. MLMS können als Resultat von 3 verschiedenen Typen von Ereignissen ausgeführt werden. Diese sind Datenbankoperationen, externe Notifikationen und zeitlich abhängige Ereignisse wie eine verzögerte oder zyklische Ausführung.

Das am häufigsten verwendete Ereignis ist dabei ein Ereignis von der Datenbank, welches durch neue Werte oder Änderungen an der Datenbank ausgelöst wird. Ein Beispiel ist das in Abbildung 2 verwendete „blood_count_storage“ Ereignis, das immer dann ausgelöst wird wenn ein neuer Wert für ein Blutbild in die Datenbank eingetragen wird.

Der Logic Slot :

Der Logic Slot enthält die eigentlichen Entscheidungsregeln des MLMs und beschreibt die medizinische Logik als einen Algorithmus. Der Logic Slot wird dabei durch ein „conclude“ Ausdruck, welcher „true“(positiv) oder „false“(negativ) zurückgeben kann, beendet.

Die im Data Slot erstellten Variablen dienen dabei der Entscheidungsfindung mithilfe von if-then-else Anweisungen.

Der Action Slot :

Der Action Slot wird ausgeführt wenn die Logik mithilfe der ausgelesenen Daten ein positives Ergebnis liefert, also der „conclude“ Ausdruck „true“ zurückgibt. Ergebnisse können in die Datenbank geschrieben werden, als Mails versendet oder als Nachrichten wie in Abbildung 2 auf dem Bildschirm angezeigt werden.

Der Language Slot:

Dieser Slot bietet Möglichkeiten zum hinterlegen von verschiedenen Sprachen um die Logik der Module von sprachabhängigen Texten zu trennen, wie z.B. Nachrichten die auf dem Bildschirm ausgegeben werden. Somit kann das MLM sprachunabhängig gemacht werden und länderübergreifend eingesetzt werden.

Operatoren/Objekte

Operatoren

Ein Schwerpunkt der Arden Syntax ist die Handhabung von Listen, Vergleichen und zeitlichen Bedingungen. In Arden werden dazu eine Reihe von Operatoren verwendet um mit Listen arbeiten zu können. Diese Operatoren umfassen die Handhabung von einfachen Listen bis hin zu komplexen Vergleichen und Rechenoperationen auf Listen. Ebenso wird die Arbeit auf Objekten unterstützt. Dazu existieren spezielle Operatoren für zeitliche Vergleiche und JOIN Operatoren auf Tabellen.

Objekte

Eine wichtige Weiterentwicklung der Arden Syntax sind die Erweiterungen an Objekten (seit Version 2.9 (Jahr 2012) [54]), die es erlauben, komplexere Datenstrukturen zu modellieren zu bearbeiten oder auszutauschen. Ebenso können über die Resources MLMs über Landesgrenzen hinweg mit verschiedenen Sprachen lokalisiert werden. Dazu meldet das System zum Zeitpunkt der ausführend des MLMs die aktuelle Systemsprache.

Datenzugriff

Mithilfe von „Curly Braces“ Ausdrücken, also Ausdrücken die in geschweiften Klammern geschrieben werden, kann das MLM direkt auf externe Datenquellen der Umgebung wie Datenbanken zugreifen.

Arden trennt dadurch explizite Referenzen an die Umgebung mithilfe von diesen „Curly Braces“ Ausdrücken von der eigentlichen Logik des MLM.

Dabei werden diese Ausdrücke an das umgebende System eins zu eins weitergeleitet. Diese werden dann von dem umgebenden System interpretiert und die angeforderten Daten zurückgeliefert. Meistens enthalten diese Ausdrücke Operationen auf Datenbanken, wie Abfragen auf Tabellen.

Das Curly Braces Problem

Das Problem ist die Anpassung der MLMs an die Zielumgebung um deren Datenquellen und deren Struktur zu entsprechen. Die HL7 Workgroup arbeitet seit einiger Zeit mit Arden 3.0 an der Lösung des Problems, bisher liegen aber noch keine Ergebnisse vor.

Das Problem ist, dass keine festgeschriebene Syntax für die „Curly Braces“ Ausdrücke existiert. Bei der Erstellung eines MLMs werden notwendige „Curly Braces“ Ausdrücke meistens im Hinblick auf die Zielumgebung und deren Datenbankschemata entworfen. Dies erschwert einen interinstitutionellen Austausch von einzelnen MLMs oder ganzen Wissensbasen.

Somit ist ein reibungsloser Austausch von MLMs nur möglich, wenn dort nur Wissen modelliert ist oder die Datenbankschemata übereinstimmen.

Implementierung von Arden

MLMs benötigen zusätzlich eine Programm, dem alle auf dem System verfügbaren MLMs bekannt sind, sodass es bei Eintreten eines bestimmten Ereignisses die betroffenen MLMs starten kann. Zusätzlich muss ein MLM zum Zeitpunkt der Ausführung in der Lage sein Informationen mit dem umgebenden System austauschen zu können.

MLMs können aus diesem Grund nicht direkt in ausführbare Programme übersetzt werden, das bedeutet, dass ein MLM entweder direkt zur Laufzeit interpretiert oder in eine Zwischensprache übersetzt werden muss.

Die Übersetzung in eine Zwischensprache hat den Vorteil, dass die MLMs auf Richtigkeit des Syntax geprüft werden, da sie von einem Compiler umgewandelt werden müssen, ebenso sind notwendige Informationen für die Ausführung wie Ereignisse schnell verfügbar.

Werkzeugunterstützung

Als Grundlage für Ruleengine-Softwares wurde die Arden-Syntax in einer Reihe von kommerziellen Produkten von Unternehmen wie Agfa-Gavaert, N.V., Allscripts, McKesson Corp, Siemens AG, Medexer Healthcare implementiert.

Neben diesen Closed-Source-Produkten existieren 2 dokumentierte Implementierungen von Arden Syntax Compilern : Arden/J und Arden2bytecode [7]. Diese ermöglichen mittels eines Compilers eine direkte Ausführung von Arden MLMs in einer Java VM. Ein Eclipse Plugin (Arden4Eclipse) ist ebenso für die Entwicklung als erweiterter Texteditor von Arden MLMs verfügbar [7].

Aktuelle Entwicklungen

Die Arden Syntax wird von verschiedenen Institutionen verwendet [45]:

Universitäre Einrichtungen:

- Cedars-Sinai Medical Center, CA
- CHU Henri Mordor, Paris
- Linköping Universität
- Universität Giessen

Softwareanbieter :

- GWI AG
- Siemens AG
- Eclipsys
- McKesson

Wissensanbieter :

- Micromedex, USA
- Medexer, Österreich

Aktuell arbeitet HL7 mit Arden 3.0 auch an der Fuzzifizierung des Arden Syntax [54], da im klinischen Alltag nicht immer mit scharfen Grenzwerten gearbeitet werden kann. Zum

Beispiel kann sich die Behandlungsdauer durch zusätzliche Faktoren von Patient zu Patient unterscheiden. In Folge dessen soll auch das Curly Braces Problem behoben werden indem das Reference Information Model von HL7 zum Datenaustausch verwendet werden soll. [54]

Beispiel von Medexter

Der Medexter Arden-Syntax-Server ist eine um SOA-Komponenten erweiterte Arden-Syntax-Ruleengine, die durch einfache Webserviceschnittstellen in beliebige Systeme integriert werden kann. Dabei sind Anpassungen an die Systemumgebung notwendig, die sich aber nur auf die Datenzugriffskomponenten beschränken [54]. Ein Betrieb ohne Datenzugriffskomponenten ist ebenso möglich, sofern die Daten beim Aufruf des MLMs übergeben werden. Im Rahmen einer Neuinstallation eines Krankenhausinformationssystems am AKH Wien wurde 2012 [54] in Zusammenarbeit von Medexter Healthcare mit der Siemens AG Softwarekomponenten zur Entwicklung und dem Einsatz von Systemen zur klinischen Entscheidungsunterstützung integriert. Dazu wurden von der Siemens AG eine Entwicklungs- und Testsoftware (Arden-Syntax-IDE(kommerziell)) beigefügt. Diese Entwicklungsumgebung ermöglicht die einfache Erstellung von Arden MLMs. Diese MLMs werden in einem weiteren Schritt von der Qualitätssicherung überprüft und dann freigegeben. Die freigegebenen MLMs werden dann auf dem Arden Syntax Server für die spätere Verwendung gespeichert und können so über ein Webinterface vom Krankenhauseigenen Informationssystem aufgerufen und ausgeführt werden.

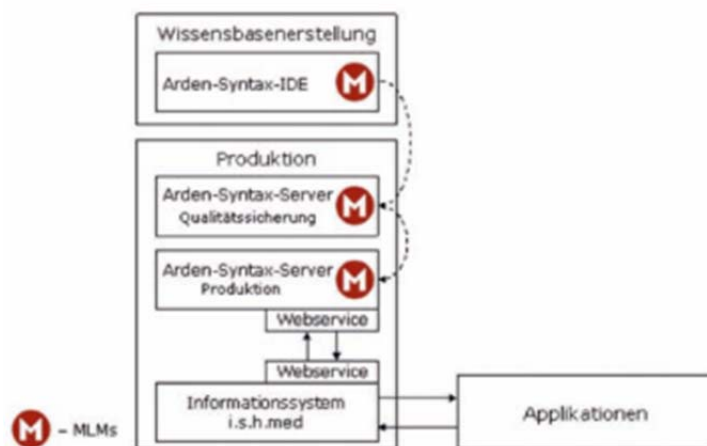


Abbildung 5 : Architektur des Softwaresystems AKH Wien 2012 [54]

3.4.2 Asbru / Asgaard

Historie

Asbru ist ein Forschungsprojekt genannt „ASGAARD Project“ aus dem Jahr 1998 der TU-Wien und der Ben Gurion University. 2002 beteiligte sich eine Fachgruppe von HL7 mit an der Entwicklung von Asbru. Asbru wurde bis zum Jahr 2006 weiterentwickelt und befindet sich derzeit in Version 7.4. Die letzte Aktualisierung der Projektseite erfolgte Juni 2006 [105].

Idee/Ziel

Ziel von Asbru ist die unscharfe Darstellung von Behandlungsabläufen um dem Anwender Freiraum für eigene Entscheidungen zu lassen.

Asbru nutzt dazu sogenannte zeitorientierte skeletale Pläne zur Repräsentation seiner Formalismen und Absichten und speichert diese im XML Format.

In ASBRU können mittels Intervallsgrenzen und eines Referenzzeitpunktes unsichere Zeitangaben gemacht werden. Beispielhaft sei hierfür eine minimale und maximale Behandlungsdauer eines Behandlungsschrittes genannt.

Diese Ungenauigkeiten werden dann im Laufe der Behandlung immer weiter verfeinert , wenn sie auf einen bestimmten Patienten angewendet werden. Zusätzlich werden in Asbru Absichten definiert, die beschreiben, welches Ergebnis eine Behandlung haben soll.

Asbru wurde zur Repräsentation von klinischen Pfaden entwickelt, nicht Leitlinien, Asbru besitzt aber dennoch die Möglichkeit Leitlinien zu repräsentieren wie von [95] bewiesen wurde.

Asbru erweitert diese skeletalen Pläne um 1.) Planattribute wie z.B. Absichten, Bedingungen, Effekte, unsichere Zeitrepräsentationen, 2.) um Operatoren für ungeordnete, sequenzielle, parallele und zyklische Ausführung von Asbru-Plänen und Aktionen.

Ein wichtiges Grundkonzept von Asbru ist es Absichten als Teil einer Behandlung zu definieren, so können die beabsichtigten Ziele einer Behandlung nach der Ausführung eines Planes überprüft werden.

Asbru wurde spezifisch für Planungsverwaltungsaufgaben entwickelt. Dies ermöglicht dem Designer sowohl die Repräsentation von vorgegebenen Aktionen eines skeletalen Plans als auch die Repräsentation von Wissensrollen (knowledge roles) für Problemlösungsmethoden und deren verflochtenen Subtasks.

Die wichtigsten Merkmale von Asbru sind [44]:

- Vorgeschriebene Handlungen und Zustände können kontinuierlich sein
- Absichten, Umstände und Weltzustände sind zeitliche Muster, d.h. sie werden alle mit Angaben von Zeiträumen beschrieben
- Die Unsicherheit kann im zeitlichen Bereich und als Parameter flexibel mithilfe von Grenzwerten ausgedrückt werden
- Pläne können in Reihe ausgeführt werden, alle oder einige Pläne parallel, alle Pläne oder einige Pläne in einer bestimmten Reihenfolge oder periodisch
- Besondere Bedingungen und Operatoren werden definiert um die Planausführung überwachen zu können
- Explizite Absichten und Einstellungen können für jeden Plan gesondert ausgewiesen werden

Sprachelemente

Sprachelemente von Asbru sind **Präferenzen, Absichten, Bedingungen, Effekte** und der **Plankorpus**.

Analog zur Klasse-Objekt Beziehung der objektorientierten Programmierung kann ein Asbru Plan als Vorlage angesehen werden. Diese Vorlage wird instanziiert, wenn ein Plan ausgeführt wird. Ebenso können mehrere Instanzen eines Planes erstellt werden.

Zeitrepräsentation

Asbru besitzt umfangreiche Möglichkeiten der Zeitrepräsentation. Die Zeitrepräsentation erfolgt immer mithilfe eines Referenzzeitpunktes „REFERENCE“ und dazu gehörigen

Intervallsgrenzen (ESS,LSS,EFS,LFS). So ist es möglich präzise Zeitangaben sowie unsichere Zeitangaben wie Anfangs- oder Endzeit oder die geschätzte Dauer eines Behandlungsschrittes darzustellen[5].

Beispiel der Zeitrepräsentation: ([ESS,LSS],[EFS,LFS],[MinDu,MaxDu], REFERENCE)

- ESS : frühest möglicher Startzeitpunkt
- LSS : spätest möglicher Startzeitpunkt
- EFS : frühest möglicher Endzeitpunkt
- LFS : spätest möglicher Endzeitpunkt
- MinDu : minimale Dauer
- MaxDu : maximale Dauer
- REFERENCE : Referenzzeitpunkt auf den sich die Zeitintervalle darstellen

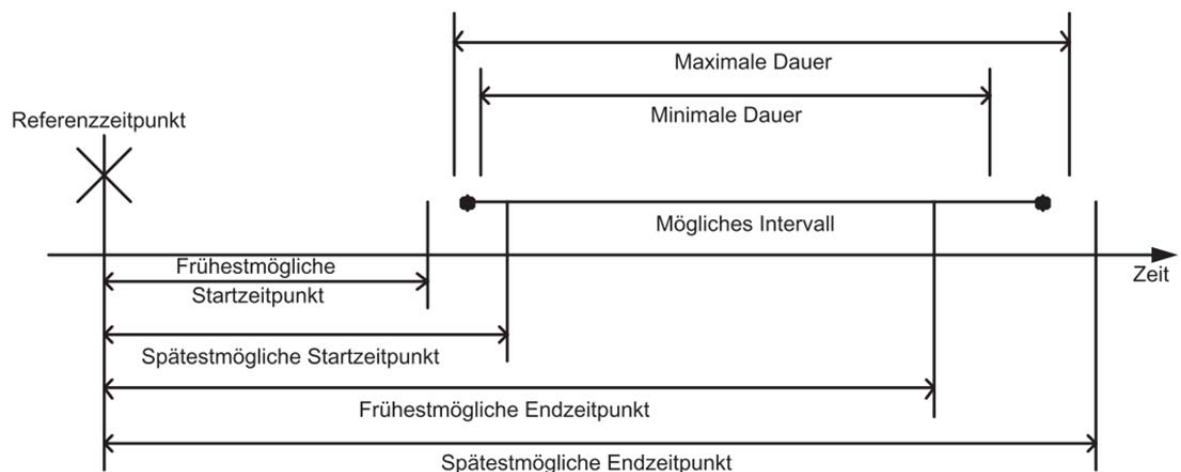


Abbildung 6: Zeitrepräsentation von Asbru

Zustände

Jeder Plan in Asbru befindet sich vor und während der Ausführung wie ein deterministischer Automat in einem definierten Zustand (Siehe Abbildung 6). Übergänge von einem Planzustand zu einem anderen werden durch Bedingungen (Conditions) definiert. Diese Bedingungen werden für jeden Zustandsübergang definiert. Dabei unterscheidet Asbru zusätzlich, ob ein Plan gerade ausgeführt wird oder sich in der Auswahlphase vor der Ausführung befindet.

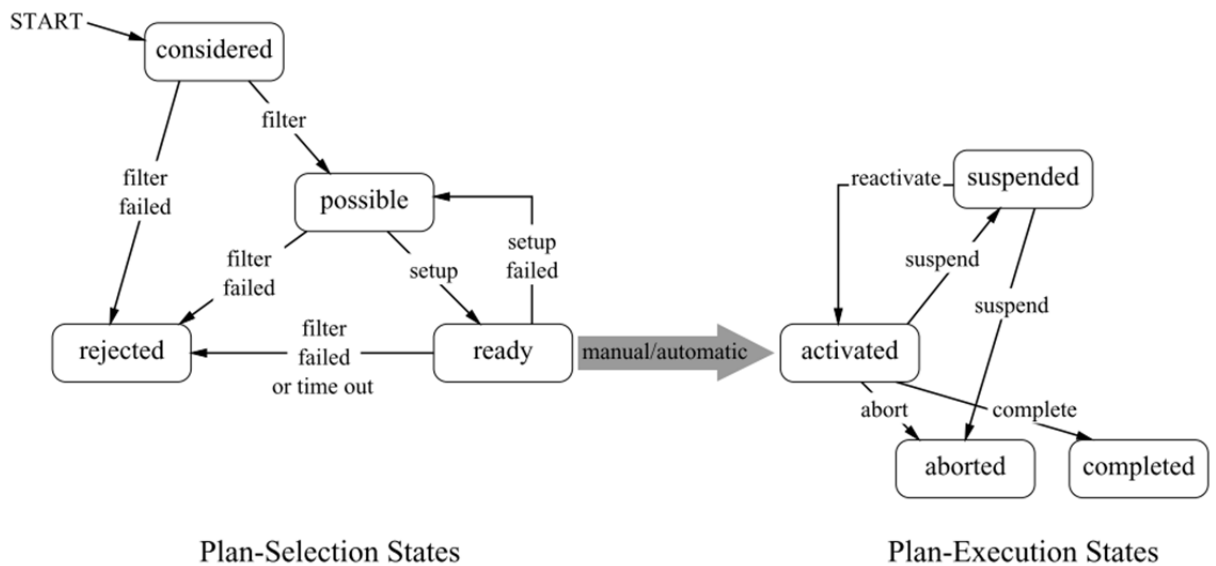


Abbildung 7 : Planzustände von Asbru

Beispielhafter Ablauf:

Nach dem Start werden die möglichen Behandlungspfade gesucht (Zustand „considered“), überprüft und gefiltert (filter condition). Die Ergebnisse der Suche sind mögliche Arten der Behandlung (possible). Aus diesen Ergebnissen kann nun eine Behandlung gewählt werden. Wenn die ausgewählte Behandlung nun notwendige Daten für die Ausführung erhält (setup condition) so kann sie ausgeführt werden (Zustand „ready“). Die Aktivierung eines eingestellten Planes erfolgt dann entweder manuell vom Benutzer oder automatisch vom System. Die Planzustände werden im Teil [Bedingungen](#) und [Präferenzen](#) noch genauer behandelt.

Absichten (Intentions)

Asbru erlaubt es Absichten als Teil einer Behandlung zu definieren, so können die beabsichtigten Ziele einer Behandlung nach Ausführung von Plänen überprüft werden. Ebenso besteht die Möglichkeit Alternativaktionen einzuleiten.

Vier Kategorien von Absichten können unterschieden werden:

- **Intermediate state** : Zustand der während der Ausführung erreicht, erhalten oder vermieden werden soll (Bsp: Zunahme oder Abnahme von Gewicht während der Behandlung)
- **Intermediate action**: Aktion die während der Ausführung des Plans durchgeführt werden soll. (Bsp: Blutzuckerspiegel messen)

- **Overall state Pattern** : Muster von Zuständen oder von Messdaten, welche nach Ausführung des Planes erfüllt sein sollen (Bsp: Patient hatte weniger als einmal zu hohe Blutzuckerwerte in der letzten Woche)
- **Overall action pattern**: Muster von Aktionen welche nach Ausführung des Planes beendet sein sollen. (Bsp: Patient macht seit mindestens 3 Monaten Diät)

Beispiel : Im folgenden Beispiel sind die Absichten des Planes , a) dass die Temperatur nicht mehr als eine Stunde über 77 Grad Celsius liegen darf und b) sollte die Temperatur immer über 50 Grad Celsius gehalten werden, nachdem der Plan 10 Minuten aktiv ist.

```

<intentions>
  <intention type="intermediate-state" verb="avoid">
    <parameter-proposition parameter-name="temperature">
      <value-description type="greater-than">
        <numerical-constant unit="C" value="77"/>
      </value-description>
      <context>
        <context-ref name="heating"/>
      </context>
      <time-annotation>
        <time-range>
          <duration>
            <minimum>
              <numerical-constant unit="h" value="1"/>
            </minimum>
          </duration>
        </time-range>
        <self/>
      </time-annotation>
    </parameter-proposition>
  </intention>
  <intention type="overall-state" verb="maintain">
    <parameter-proposition parameter-name="temperature">
      <value-description type="greater-than">
        <numerical-constant unit="C" value="50"/>
      </value-description>
      <context>
        <context-ref name="heating"/>
      </context>
      <time-annotation>
        <time-range>
          <finishing-shift>
            <latest>
              <numerical-constant unit="min" value="10"/>
            </latest>
          </finishing-shift>
        </time-range>
        <self/>
      </time-annotation>

    </parameter-proposition>
  </intention>
</intentions>

```

Abbildung 8 : Intention Beispiel aus der Asbru Reference Manual [44]

Bedingungen (Conditions)

Bedingungen definieren die Transition von Planzuständen. Erst wenn diese Bedingungen erfüllt sind kann ein Asbru Plan einen Zustandsübergang in einen anderen Planzustand ausführen. Diese Bedingungen sind optional, d.h. wird keine Bedingung definiert so kann der Zustandsübergang immer ausgeführt werden. Folgende 6 Typen von Bedingungen existieren:

- **Filter-preconditions** : Diese Bedingungen müssen erfüllt sein, damit der Plan gestartet werden kann. Die Bedingungen können dabei aber nicht erreicht werden (z.B. Patient ist ein Neugeborener)
- **Setup-preconditions**: Diese Bedingungen müssen erreicht werden damit der Plan gestartet werden kann (z.B. Gesundheitszustand des Patienten ist schlecht)
- **Suspend-conditions**: Gibt Bedingungen an, unter denen ein Plan pausiert werden muss.
- **Abort-conditions** : Gibt Bedingungen an, unter denen ein gestarteter, wiederaufgenommener oder ausgesetzter Plan abgebrochen werden muss (z.B. Blutgaswerte können bei kontrollierter Beatmung nicht stabilisiert werden)
- **Complete-conditions** : Gibt Bedingungen an, unter denen ein gestarteter oder wieder ausgeführter Plan beendet werden kann
- **Restart-conditions** : Geben an, wann ein ausgesetzter Plan wieder aufgenommen werden kann (z.B. Hämoglobinwert im Blut wieder über 10mg/ml)

Beispiel : Das folgende Beispiel zeigt alle möglichen Bedingungen eines Asbru Planes. Der Plan wird in diesem Beispiel automatisch gestartet und erfordert keinen Benutzer zum Starten. Da keine weiteren Bedingungen definiert sind, wird dieser Plan direkt nachdem er ausgewählt wurde gestartet.

```

<conditions>
  <filter-precondition>
    ...
  </filter-precondition>
  <setup-precondition>
    ...
  </setup-precondition>
  <activate-mode started="automatic"/>
  <suspend-condition>
    ...
  </suspend-condition>
  <reactivate-condition>
    ...
  </reactivate-condition>
  <complete-condition>
    ...
  </complete-condition>
  <abort-condition>
    ...
  </abort-condition>
</conditions>

```

Abbildung 9 : Beispiel von Bedingungen in Asbru aus der Asbru Reference Manual [44]

Präferenzen

Präferenzen beschränken und beeinflussen die Auswahl von Plänen. Sie beschreiben wie ein Plan ausgeführt werden soll, d.h. welche zusätzlichen Ziele , abgesehen von der Gesundheit des Patienten, mit der Behandlung verfolgt werden sollen.

Unterschieden wird zwischen:

- **Strategie** : Allgemeine Strategie des Planes (aggressive ,normale Behandlung)
- **Zweck** : Eine Menge von Zweckangaben (z.B. minimieren der Kosten)
- **Auswahlmethode** : Auswahlheuristik für die Verwendbarkeit des Planes. Asbru nutzt dabei eine Vielzahl von selbst definierten Operatoren. Diese Operatoren beschreiben wann genau ein Plan zur Verwendung kommen kann , z.B. „exact-fit“ beschreibt, dass alle Bedingungen genau erfüllt sein müssen um den Plan ausführen zu können, wobei „roughly-fit“ beschreibt, dass nicht alle Bedingungen erfüllt sein müssen um einen Plan auszuführen
- **Ressourcen** : Spezifikation von verbotenen und vorgeschrieben Ressourcen(z.B. Ausschluss/Verbot von bestimmten Medikamenten oder Geräten)
- **Startbedingungen**: Gibt an, ob der Plan vom Benutzer oder automatisch instanziiert werden kann.

Beispiel: Das folgende Beispiel sagt aus, dass der Plan ein Gerät namens „Device-A“ für eine Behandlung von 1 bis 2 Stunden benötigt. Die voraussichtlichen Kosten werden 1000 Euro betragen und die Nebenwirkungen beim Patienten werden gering sein. Des Weiteren wird die Strategie des Planes als konservativ eingestuft und der Patient ist für die Aufnahme der Maßnahmen zuständig.

```
|<preferences responsible-actor="patient" strategy="conservative">
  <resource-constraint name="device-A" type="obligatory">
    <time-annotation>
      <time-range>
        <duration>
          <minimum>
            <numerical-constant unit="h" value="1"/>
          </minimum>
          <maximum>
            <numerical-constant unit="h" value="2"/>
          </maximum>
        </duration>
      </time-range>
    </self/>
  </time-annotation>
</resource-constraint>
<costs name="monetary-costs">
  <numerical-constant unit="Euro" value="1000"/>
</costs>
<costs name="discomfort">
  <qualitative-constant value="low"/>
</costs>
</preferences>
```

Abbildung 10 : Preference Beispiel aus der Asbru 7 Reference Manual [44]

Effekte

Effekte beschreiben die quantitativen Auswirkungen von Plänen unabhängig von Bedingungen und Absichten. Sie beschreiben welche „Effekte“ die Behandlung auf den Patienten haben müsste. Ebenso können Wahrscheinlichkeitswerte für das Auftreten von Effekten spezifiziert werden.

Beispiel: Im Kontext der Behandlung von Schwangerschaftsdiabetes (GDM) kann die Behandlung eine Senkung des Blutzuckerspiegels mit einer Wahrscheinlichkeit von 0.97 (97%) zur Folge haben. Bei der Behandlung dauert es 10 bis 60 Minuten bis der Effekt auftritt.


```

<plan-effect direction="dec" likelihood="0.97" parameter="glucose-level">
  <time-annotation>
    <time-range>
      <starting-shift>
        <earliest>
          <numerical-constant unit="min" value="10"/>
        </earliest>
        <latest>
          <numerical-constant unit="min" value="60"/>
        </latest>
      </starting-shift>
    </time-range>
    <self/>
  </time-annotation>
  <context>
    <context-ref name="gdm"/>
  </context>
</plan-effect>

```

Abbildung 11 : Asbru Effect Beispiel aus der Asbru Reference Manual [44]

Plankorpus (Plan-Library)

Pläne dienen in Asbru der Strukturierung und Kategorisierung von Behandlungspfaden.

Plankorpus und dessen Struktur

Der Plankorpus (Plan-Library) enthält die auszuführenden Aktionen und Subpläne, sowie Definitionen von Variablen, Konstanten und Iteratoren die später von den Plänen genutzt werden. Spezielle in Asbru definierte Operatoren ermöglichen die Spezifikation von ungeordneter, sequentieller, paralleler und zyklischer Ausführung von Plänen oder Aktionen. Wichtige Elemente sind dabei der **Subplan**, **cyclical plan** und **single-step**.

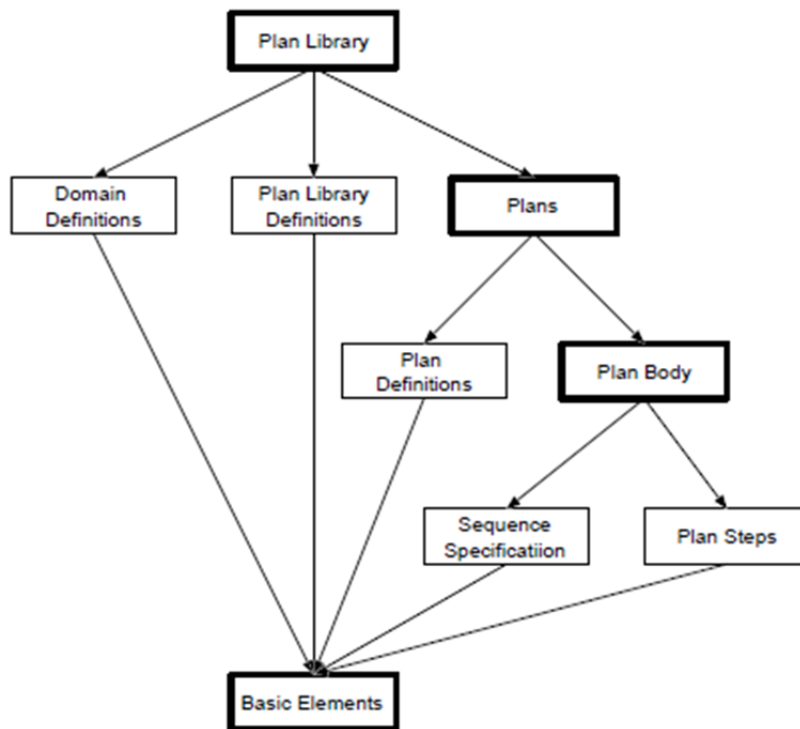


Abbildung 12 : XML Ausbau eines Asbru Planes nach Reference Manual [44]

Schematischer Aufbau eines Asbru XML Planes : Wurzelement ist in jedem Asbru Plan das „Plan Library“. Dick markierte Rechtecke korrespondieren zu den vorgestellten Sektionen. Diese enthalten wiederum Pläne (Plans) mit unterschiedlichen Aktionen und Absichten sowie Effekten. Innerhalb des Plan-Body werden diese dann beschrieben.

Beispiel : Das folgende Beispiel zeigt Fragmente des Plankorpus um dessen Struktur zu zeigen. Ebenso werden hier einige Variablen, Konstanten und Iteratoren definiert, die später von den Plänen verwendet werden könnten.

```

<plan-library>
  <library-info title="Skeleton of a Plan Library">
    ...
  </library-info>
  <domain-defs/>
  <library-defs>
    <variable-def name="List-1" scalar-or-not="list" type="string">
      <comment text="List-1 is a list of strings"/>
    </variable-def>
    <constant-def name="PI">
      <numerical-constant unit="amount" value="3.1415"/>
    </constant-def>
    <iterator-def name="Iterator-1">
      <variable-ref name="List-1"/>
    </iterator-def>
    <function-def class-name="asgaard.checkit"
      method-name="add_em_up"
      name="add"
      return-type="length"/>
  </library-defs>
  <plans>
    <plan-group>
      <plan name="Plan-A">
        ...
      </plan>
      <plan name="Plan-B">
        ...
      </plan>
      <plan name="Plan-C">
        ...
      </plan>
    </plan-group>
    <plan-group>
      ...
    </plan-group>
  </plans>
</plan-library>

```

Abbildung 13 : Plan-Library aufbau nach Asbru Reference Manual [44]

Elemente im Plan:

- **Cyclical-plan** : periodisch wiederholter Plan mit zusätzlichen Attributen wie z.B. die Anzahl der Wiederholungen oder der Angabe eines bestimmten
- **Single-step** : einzelner abstrakter schritt einer Planausführung wie z.B. die zuweisung eines Wertes zu einer Variable oder die Aktivierung eines anderen
- **Refer-to** : Bezug auf Plankorpus eines anderen Planes
- **To-be-defined** : gibt an das der Plan nicht ausführbar ist (abstrakt ist)
- **User-performed** : Schritt wird von einem Anwender und nicht vom System ausgeführt

- **Subplans** : Sind eine Menge von sequentiell oder parallelen Planschritten die gruppierte Abläufe definieren. Diese können folgende Eigenschaften besitzen:
 - o Sequential : Die Schritte werden nacheinander ausgeführt

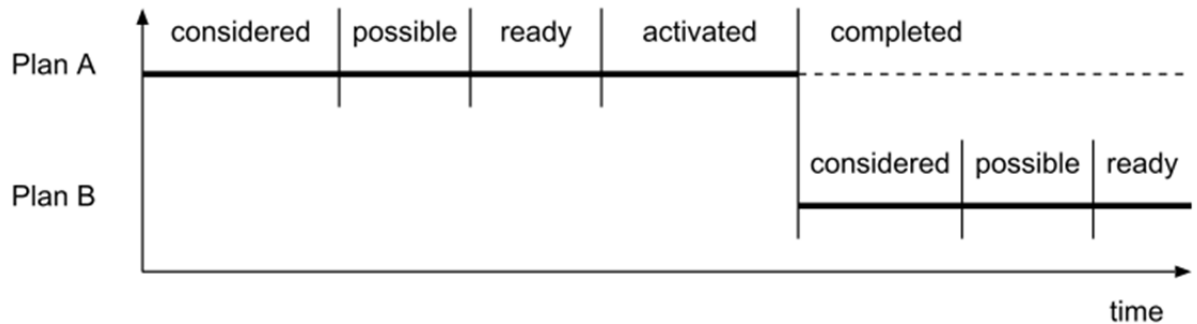


Abbildung 14 : Beispielvisualisierung eines sequentiellen Plans

- o Parallel: Alle Pläne des subplanes werden gleichzeitig gestartet. Sie können sich zeitlich unabhängig voneinander beenden. Jeder Plan wartet dabei vor einem Zustandsübergang auf alle parallel laufenden Pläne

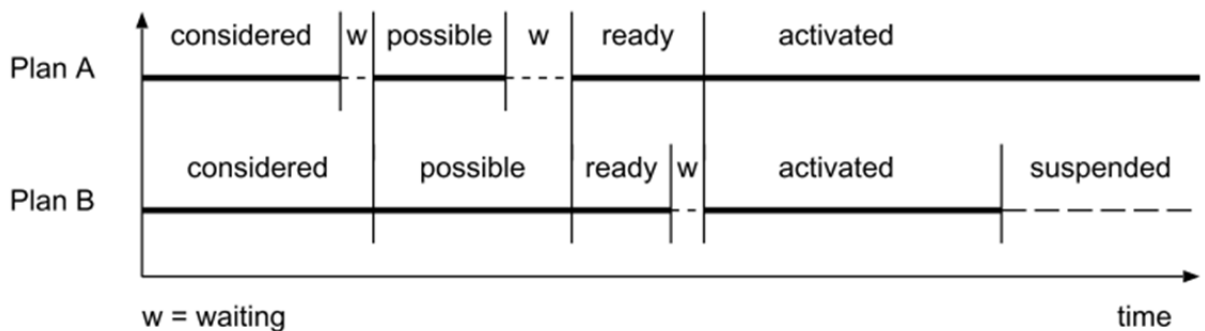


Abbildung 15 : Beispielvisualisierung eines parallelen Plans

- o Any-order: Subpläne werden in unbestimmter Reihenfolge ausgeführt. Dabei ist aber immer nur ein Subplan in Ausführung.

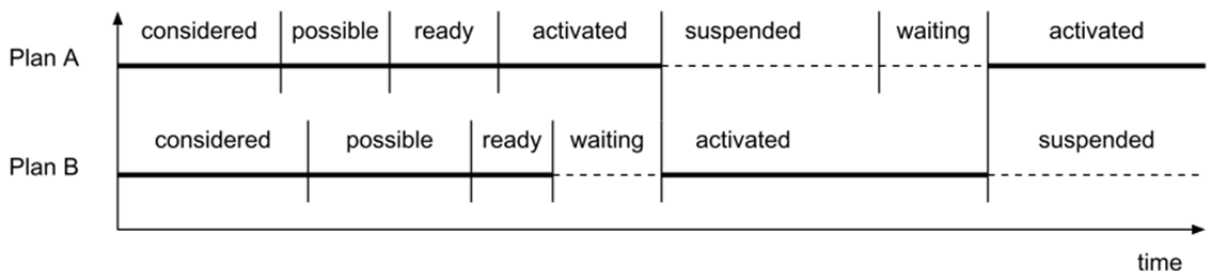


Abbildung 16 : Beispielvisualisierung eines Any-order Plans

- unordered : es gibt keine Ordnung zwischen den Schritten, sie können sich überlappen oder auch nicht. Jeder Schritt wird dann ausgeführt wenn es passend ist

Beispiel: Das folgende Beispiel zeigt Plan „A“, der jede Stunde aufgerufen wird. Seine typische Dauer beträgt 30min. Plan „B“ wird aktiviert wenn Plan „A“ Abbricht und Plan „C“ wird aktiviert wenn Plan „A“ angehalten (suspended) wird.

```
<plan name="A">
  <defaults>
    <on-suspend>
      <plan-activation>
        <plan-schema name="C"/>
      </plan-activation>
    </on-suspend>
    <on-abort>
      <plan-activation>
        <plan-schema name="B"/>
      </plan-activation>
    </on-abort>
    <time-annotation>
      <time-range>
        <duration>
          <maximum>
            <numerical-constant unit="h" value="1"/>
          </maximum>
        </duration>
      </time-range>
      <self/>
    </time-annotation>
    <typical-duration>
      <numerical-constant unit="min" value="30"/>
    </typical-duration>
  </defaults>
</plan>
```

Abbildung 17 : Planbeispiel von Asbru aus der Asbru Reference Manual [44]

Werkzeuge

Es existieren eine Reihe von Werkzeugen zur Darstellung und Entwicklung von klinischen Pfaden in Asbru. Von der Technischen Universität Wien werden dazu die Tools AsbruView [11] und DELT/A, welches sich auf die XML Dokumentbearbeitung konzentriert [107] angeboten. Diese Tools werden seit 2006 nicht mehr weiterentwickelt. Stepper ist ebenso ein Werkzeug zur Erstellung von klinischen Pfaden und wird auch nicht mehr weiterentwickelt [112].

Es werden keine Vorgehensbeschreibungen zur Entwicklung von Behandlungen in die Asbru Syntax bereitgestellt.

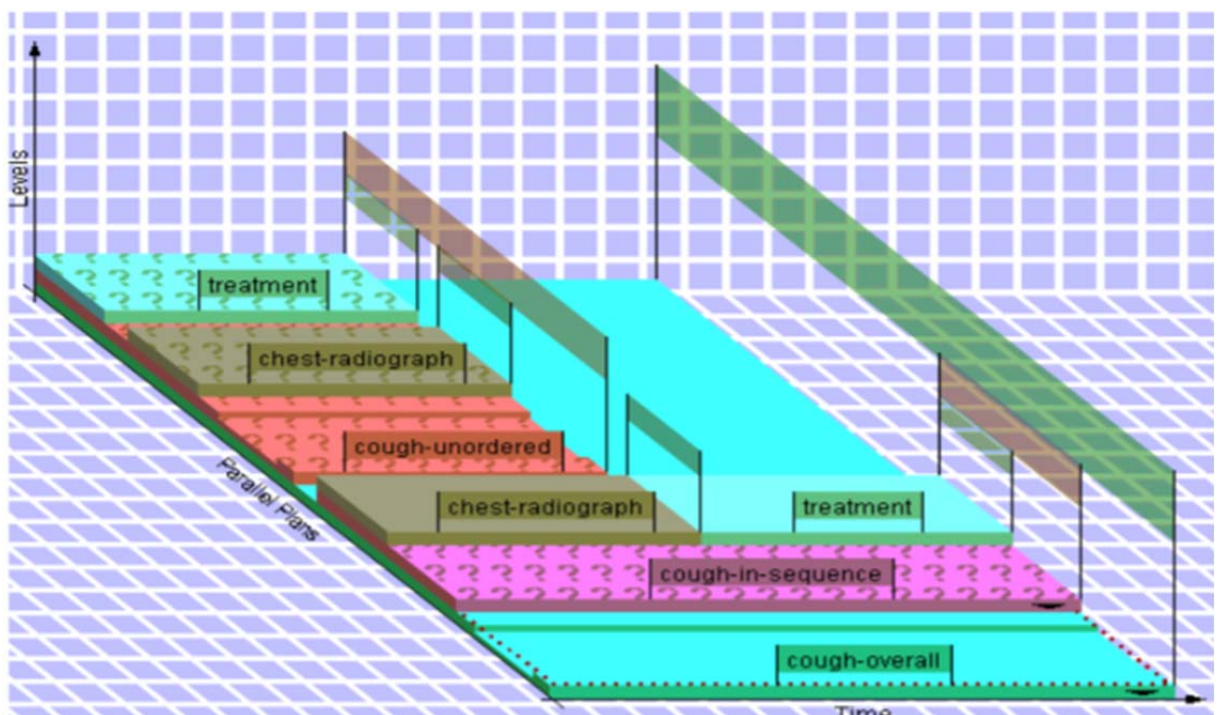


Abbildung 18 : ASBRUView Beispielhafte Darstellung der Behandlung von chronischem Husten von [80]

Die Obige Abbildung zeigt 2 verschiedene Varianten der Behandlung von chronischem Husten. Der obere Teil des Asbru Planes stellt die ungeordnete Behandlung von chronischem Husten dar. Der Untere Teil zeigt die sequentielle Bearbeitung von chronischem Husten, wobei zu erkennen ist, dass die Behandlungsdauer ansteigt, da alle Schritte nacheinander ausgeführt werden.

Aktuelle Entwicklungen

Die letzte Publikation mit dem Thema Asbru der TU Wien erfolgte im Jahr 2009 [2]. Die Asbru Syntax wurde dabei zuletzt 2006 auf Version 7.4 aktualisiert [10]. Kommerzielle Werkzeuge basierend auf ASBRU lassen sich nicht finden.

3.4.3 Klinischer Algorithmus

Historie

Der klinische Algorithmus wurde in den 60er Jahren entwickelt und wird seitdem bei der Modellierung von Schlüsselempfehlungen in der Medizin eingesetzt [93].

Idee/ Ziel

Schlüsselempfehlungen sind Empfehlungen einer medizinischen Leitlinie, die wesentliche medizinische Fragen enthalten, z.B. wie ein bestimmter Behandlungsschritt zu erfolgen hat. Für S3 Leitlinien gilt, dass die Logik von Schlüsselempfehlungen der Leitlinie algorithmisch dargestellt sein sollte [3]. Die Arbeitsgemeinschaft der Wissenschaftlichen Medizinischen Fachgesellschaften (AMWF) sowie das Ärztliche Zentrum für Qualität in der Medizin (ÄZQ) empfehlen den Einsatz von klinischen Algorithmen zur Darstellung dieser Schlüsselempfehlungen. Dafür werden bestimmte Standartelemente empfohlen [3; 79].

Die Idee war, dass Schlüsselempfehlungen für den Nutzer einer Leitlinie eindeutig erkennbar und verstehbar sind.

Ziel des klinischen Algorithmus ist eine einfache an Flussdiagrammen orientierte schrittweise graphische Prozedur mit endlich vielen Anweisungen und bedingter „wenn-dann“ Logik zur Lösung eines klinischen Problems. Bis auf eine Definition des Patientenzustandes am Anfang des Algorithmus wird dieser komplett patientenunabhängig beschrieben.

Im Gegensatz zu mathematischen Algorithmen, welche eindeutige Aussagen erfordern, können beim klinischen Algorithmus unsichere Informationen genutzt werden, z.B. können Temperaturen nicht nur in numerischen Werten angegeben werden, sondern auch unsichere Angaben wie heiß oder kalt. Jedoch wird dadurch die Einschränkung gesetzt, dass der klinische Algorithmus im Gegensatz zu einem mathematischen Algorithmus nicht bei allen Problemen anwendbar ist. Klinische Algorithmen sind als Kombination von logischer Struktur und Grafik mit wenigen terminologischen Zeichen einfach zu verstehen.

Sprachelemente

Beim klinischen Algorithmus werden 4+1(optional) Sprachelemente von AMWF und ÄZQ empfohlen, die nachfolgend genauer erläutert werden [113]:

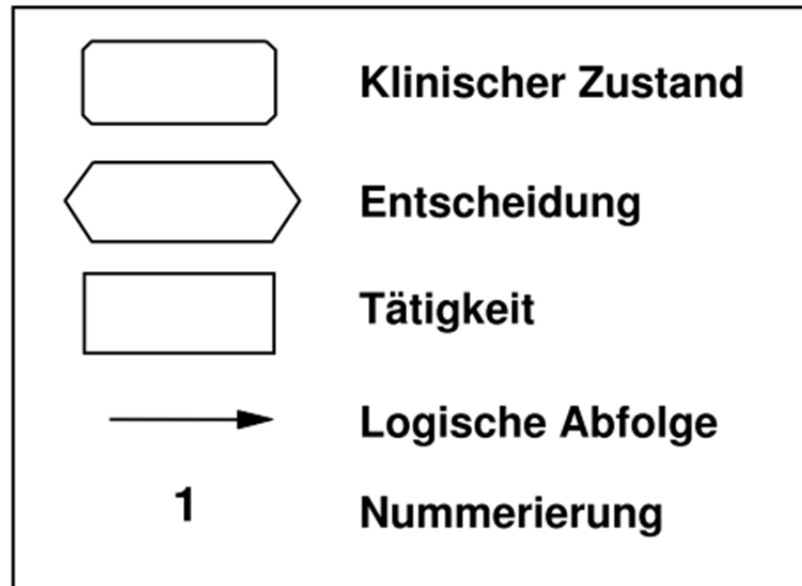


Abbildung 19 : Terminologie Klinischer Algorithmus

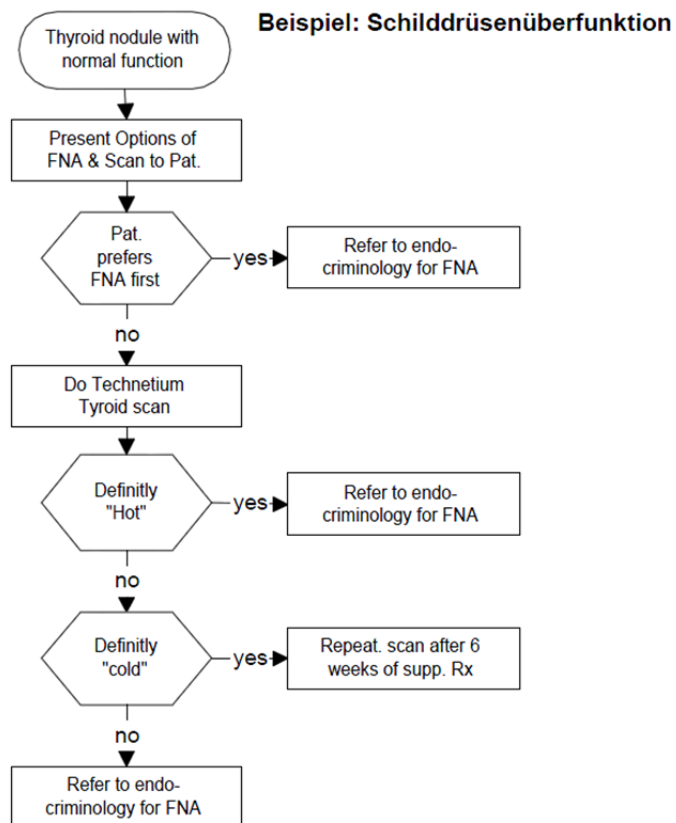


Abbildung 20 : Sehr einfaches Beispiel eines klinischen Algorithmus von [93]

Klinischer Zustand

Zu Beginn eines klinischen Algorithmus wird der Zustand des Patienten beschrieben, der die Behandlung erhalten soll. Hier werden Aussagen getroffen, die auf den jeweiligen Patienten zustimmen müssen um den Prozess auszuführen, z.B. die Annahme bestimmter Mess- oder Diagnoseergebnisse.

Entscheidung

Entscheidungen beschränken sich im klinischen Algorithmus auf Ja oder Nein. Damit ist bei einer Entscheidung immer zwischen Ja und Nein zu wählen. Andere Möglichkeiten der Entscheidung, wie einer Multiple Choice Auswahl, gibt es nicht.

Tätigkeit

Tätigkeiten eines Behandlungsprozesses werden in diesen Elementen festgehalten. Sie unterliegen keiner speziellen Syntax oder Semantik. Sie beschreiben Teiltätigkeiten eines Behandlungsprozesses innerhalb des Algorithmus.

Logische Abfolge

Mithilfe von Pfeilen wird die Abfolge von Patientenzustand, Entscheidungen und Tätigkeiten definiert und patientenunabhängig beschrieben.

Nummerierung (optional)

Für einen besseren Überblick besteht die Möglichkeit alle Einzelelemente ohne Pfeile zu nummerieren. So soll aus komplexeren Algorithmen der Ablauf schneller klar werden.

Werkzeuge

Für die Entwicklung von klinischen Algorithmen können viele generische Werkzeuge sowie Editoren genutzt werden. Zum Beispiel können klinische Algorithmen mithilfe von Microsoft Visio oder LibreOffice erstellt werden. Ebenso können generische Modelleditoren wie ARIS [8] genutzt werden.

Aktuelle Entwicklungen

Der klinische Algorithmus wurde und wird derzeit genutzt um Schlüsselempfehlungen von Leitlinien der Stufe S3 nach Empfehlung von AMWF und ÄZQ zu visualisieren.

3.4.4 EON /Dharma

Historie

EON wurde 1996 als Weiterentwicklung von ONCOCIN an der Stanford University entwickelt.

Idee/Ziel

Idee von EON war die Entwicklung einer Ontologie [91] mithilfe derer eine Vielzahl von domänenspezifischen Unterschieden in der Medizin modelliert werden können, und um patientenspezifische Entscheidungshilfen zu bieten [90].

Das Ziel von EON ist die Bereitstellung von Komponenten zur Erstellung und Ausführung von klinischen Pfaden und protokollbasierten Applikationen. EON bietet ein Standardmodell von Klassen, welches jedoch nicht auf diese Grundklassen beschränkt ist, sodass eine Erweiterung des Modells möglich ist. Mithilfe dessen können die Ontologien um Klassen und Attribute erweitert werden.

EON besteht aus einem Patienten-Informationssystem, einem medizinischen Konzeptmodell und einem taskbasierten Leitlinienmodell, Dharma Modell genannt [90]. EON selbst kann man so als eine Art Middleware für klinische Systeme sehen. EON nutzt das Dharma Modell zur Codierung seiner Leitlinien, da EON selbst keine formale Sprache hierfür bereitstellt.

Das Leitlinienmodell, Dharma-Modell genannt, ist objektorientiert und definiert abstrakte Klassen die Leitlinien mit Wissensstrukturen wie Leitlinienalgorithmen, Entscheidungsmodellen, empfohlenen Aktionen, Abstraktionsdefinitionen und Eignungskriterien definiert [61]. Benutzer können Erläuterungen einsehen mit deren Hilfe sie die generierten Empfehlungen nachvollziehen können..

Sprachelemente

Das Dharma Modell beschreibt Leitlinien als Entitäten von strukturiert zeitabhängigen Schritten, sodass eine Visualisierung als gerichteter Multigraph möglich ist.

Das Dharma Modell besteht aus Entscheidungen und Aktionen, die abhängige Veränderungen am Patientenzustand hervorrufen können. Mithilfe von ausgelesenen

Patientendaten und temporalen Abfragen können dann Empfehlungen erzeugt werden. Diese temporalen Abfragen werden dann während der Leitlinienausführung in Datenbankabfragen umgewandelt. Diese Abfragen dienen dem Erfragen von Messwerten oder Patientendaten:

```
TEMPORAL SELECT domain_name
VALID INTERSECT(Condition, Medication)
FROM Condition, Medication
WHERE domain_name = "UNCONTROLLED_BP" AND
      drug_name = "lisinopril"
WHEN start(Condition) AFTER start(Medication) AND
      start(Condition) BEFORE
      (start(Medication) + weeks(2))
```

Abbildung 21 : Beispiel einer temporalen Abfrage in EON

Dieses Beispiel einer Temporalen Abfrage prüft auf einen Zeitraum von unkontrolliertem Bluthochdruck, der mit der Verabreichung von Lisinopril überlappt und innerhalb von 2 Wochen nach Verabreichung von lisinopril auftritt.

Grundlegende Elemente des Dharma-Leitlinienmodells

Scenarios

Scenarios beschreiben die unterschiedlichen Entscheidungsmöglichkeiten (Alternativen) einer Leitlinie zur Behandlung des Patienten, die vom jeweiligen Patientenzustand abhängig sind. Scenarios beschreiben zugleich partiell den Zustand des Patienten (z.B. Patient erhält derzeit ein schwaches Steroid).

Auswählbare Bedingungen bestimmen dann mithilfe weiterer notwendiger Bedingungen ob sich ein Patient in einem bestimmten Szenario befinden kann. Scenarios werden auch als Einstiegspunkt in Leitlinien genutzt und erlauben so das Management sowie die Synchronisation mit dem korrespondierenden Teil einer Leitlinie. In Dharma folgt einem Szenario immer ein decision- oder action_step.

Jedes Szenario Objekt enthält zusätzliche Attribute wie Auswahlkriterien für nachfolgende Scenarios. Jeder alternative Behandlung einer Leitlinie ist dabei eine Instanz der scenario Klasse , welche die Auswahlkriterien und Folgeschritte beschreibt.

Scenarios können mithilfe von „exceptions“ Ausnahmen modellieren, dabei werden 2 Ausnahmetypen unterschieden:

- **repairable exceptions** : sind Ausnahmen die reparierbar sind und einen Patienten wieder zu einem szenario der Leitlinie zurückführen
- **exceptions that are not repairable** : sind Ausnahmen die nicht reparierbar sind , sodass der Patient auserhalb der Leitlinie Behandelt werden muss.

Decisions / Entscheidungen

Beschreibt eine Auswahl von Entscheidungsmöglichkeiten konkurrierender Behandlungsalternativen. In Dharma existieren grundsätzlich 2 Arten von Entscheidungen. Diese sind if-then-else Entscheidungen und heuristische Entscheidungen von vorher ausgewählten alternativen, anhand von enumerierten Variablen. Für heuristische Entscheidungen wird die Auswahl von Alternativen durch Präferenzen gestützt, die als Eingangs- und Ausgangsregeln alternativen entweder unterstützen oder ablehnen.

Ergibt eine Ausgangsregel wahr/true, dann wird diese Alternative abgelehnt. Wenn die Ausgangsregel falsch/false ergibt und die Eingangsregel wahr/true ergibt, so wird die korrespondierende Alternative als bevorzugt markiert. Evaluieren beide Regeln mit false, dann kann die Präferenz für die Auswahl von einer standardisierten und allgemeineren Präferenz ermittelt werden.

Actions / Activities

Actions werden im EON Modell stark genutzt. Sie stellen unmittelbare Aktionen dar, die mit der Umwelt agieren müssen und deren Zustand verändern (Patientendaten Messen, eine Nachricht anzeigen). Diese Aktionen werden zumeist durch die Leitlinien bestimmt.

Während Actions unmittelbare Veränderungen hervorrufen sind Activities Modelle für Prozesse, die im Laufe der Zeit stattfinden. Activities haben dafür Stati, die sich dabei von Zeit zu Zeit ändern. Diese Veränderungen sind üblicherweise ein Resultat von Actions einer Leitlinie, da diese in der Lage sind Activities zu starten, zu stoppen oder die Attribute einer laufenden Activity zu verändern.

Activitys besitzen zusätzliche Attribute wie z.B. Dosierungshöhe (wenig/viel/mittel) und Frequenz einer Aktion (z.B. Zweimal am Tag).

Activities sowie Actions können mithilfe von Referenzen auch aus mehreren Subaktionen bestehen.

Goals / Ziele

Jeder Schritt (Activity oder Action) kann mit einem Ziel in Verbindung gebracht werden. Die Modellierung von Zielen ist vergleichbar mit der Notation von Intentions/Absichten in Asbru, aber weniger anspruchsvoll. In Dharma werden Ziele als boolesche Kriterien dargestellt und können nur wahr oder falsch ergeben.

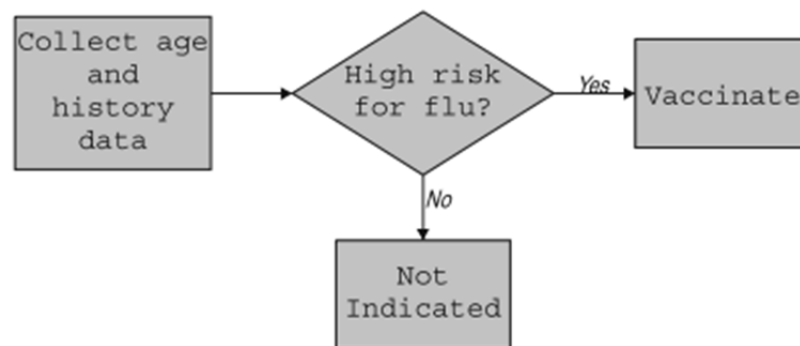


Abbildung 22 : Graphische Darstellung einer Influenza Impfung in EON

In EON werden die Leitlinien auf oberster Ebene als Graphen/Flowcharts dargestellt. Beispielhaft dargestellt am Beispiel einer Influenza Impfung. Eine Vielzahl von Klassen spielt dabei eine Rolle, können aber hier aus Platzgründen nicht dargestellt werden.

Werkzeugunterstützung

EON selbst nutzt das Modellierungswerkzeug Protegé um domänenspezifische Ontologien, ein Patienteninformationsmodell und eine Leitlinienwissensbasis zu erstellen. Zusätzlich können zu dem Kernmodell weitere abgeleitete Funktionalitäten definiert werden.

Protegé unterstützt die Visualisierung von EON-Leitlinien nur bis version 1.9 und eher, da seid Protegé 2.0 ein neues Graph Widget entwickelt wurde, welches das alte ersetzte.

Datenzugriff

EON erhält Patientendaten über einen temporalen Datenbankmanager oder Nutzereingaben um Empfehlungen basierend auf dem spezifischen Leitlinieninhalt zu generieren. Dieser Datenbankmanager ermöglicht spezielle abfragen über Zeiträume mittels einer SQL ähnlichen Syntax.

Aktuelle Entwicklungen

Im März 2003 endete die Unterstützung für das EON Projekt. Einige Resultate des Projektes wurden in das SAGE Projekt übertragen. Das SAGE Projekt ist ein auf EON basierendes Gemeinschaftsprojekt der EU welches 2006 eingestellt wurde [71]. Ebenso existiert mit HELEN 3 ein auf EON basierendes Projekt des Universitätsklinikums Heidelberg [106], welches vorgefertigte medizinische Sachverhalte bereits beinhaltet um so die Leitlinienentwicklung und deren Eindeutigkeit zu verbessern. Auf beide Projekte wurde nicht weiter eingegangen, da sich die Modellierungsaspekte zu EON nicht unterscheiden.

Auf der Homepage der Stanford University lassen sich keine Relevanten Publikationen mehr finden, die EON betreffen. Eine Projektseite existiert auch nicht mehr.

Ein kommerzielles Produkt basierend [73] auf EON ist ATHENA [109], eine inzwischen cloudbasierte kommerzielle Komplettlösung mit bereits implementierten Leitlinien für Krankenhäuser in Amerika, da die Implementierten Leitlinien den amerikanischen Gesetzgebungen entsprechen [109]. Über dieses Produkt sind nur allgemeine Informationen verfügbar, keine tiefergehenden Informationen, ob das EON Standardmodell weiter genutzt wird und wie es verwendet wird.

Ein Freies Programm war T-HELPER [92] , ein Programm der Stanford University of Medicine zum Datenmanagement von Patienten mit HIV, das Tool wird ebenso nicht mehr weiterentwickelt .

3.4.5 GLARE - Guideline Acquisition, Representation and Execution

Historie

GLARE wurde 1997 an der „Università del Piemonte Orientale (Amedeo Avogadro) Alessandria“ in Zusammenarbeit mit dem Krankenhaus von „St. Johannes dem Täufer“ in Turin und dem „Laboratorio di Informatica Clinica“ entwickelt.

Idee/Ziel

Idee von GLARE ist die Bereitstellung eines konzeptuellen Komplettsystems für klinische Pfade, um diese erstellen, darstellen und ausführen von zu können.

Ziel von GLARE ist die Bereitstellung einer benutzerfreundlichen Software die über alle medizinischen Fachrichtungen einsetzbar sein soll.

GLARE besteht dazu aus einem Darstellungsformalismus, einem Wissens-Authoring-Werkzeug und einem Ausführungswerkzeug.

Architektur von GLARE

GLARE basiert auf einer modularen Architektur, die Erstellung, Darstellung und Ausführung von klinischen Pfaden voneinander trennt. Dazu werden 2 Arten von Nutzern unterschieden. Zum einen erfolgt die Pfaderstellung in GLARE von Ärzten die sehr gute Kenntnisse von Leitlinien besitzen und diese computerinterpretierbar in formelle Abläufe codieren können. Die zweite Nutzergruppe sind die Ärzte, die das System später zur Behandlung von Patienten nutzen sollen. Folgende Darstellung zeigt ein konzeptuelles Modell des modularen Aufbaus

von GLARE.

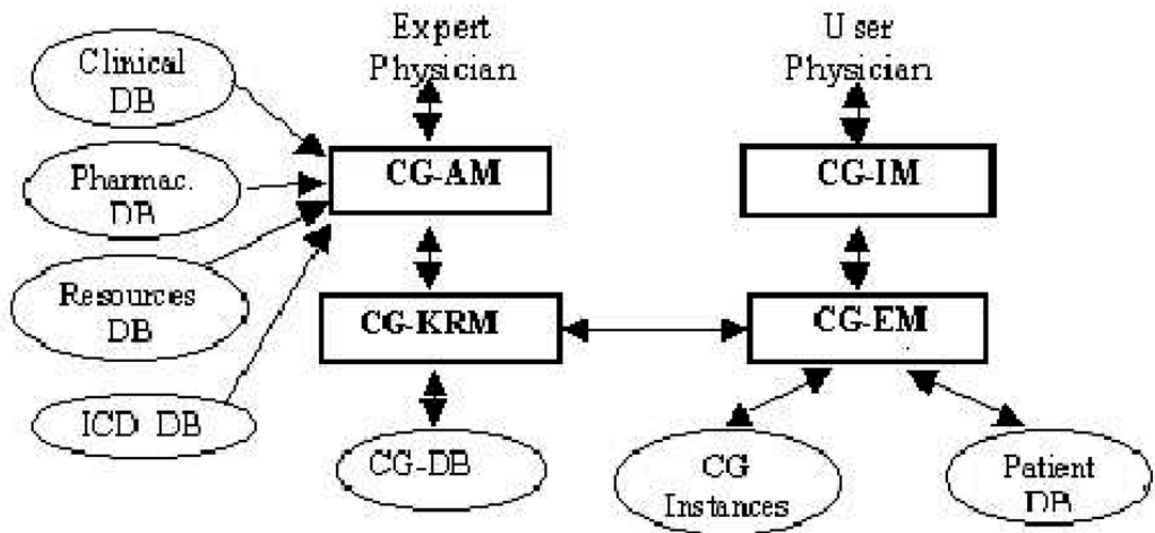
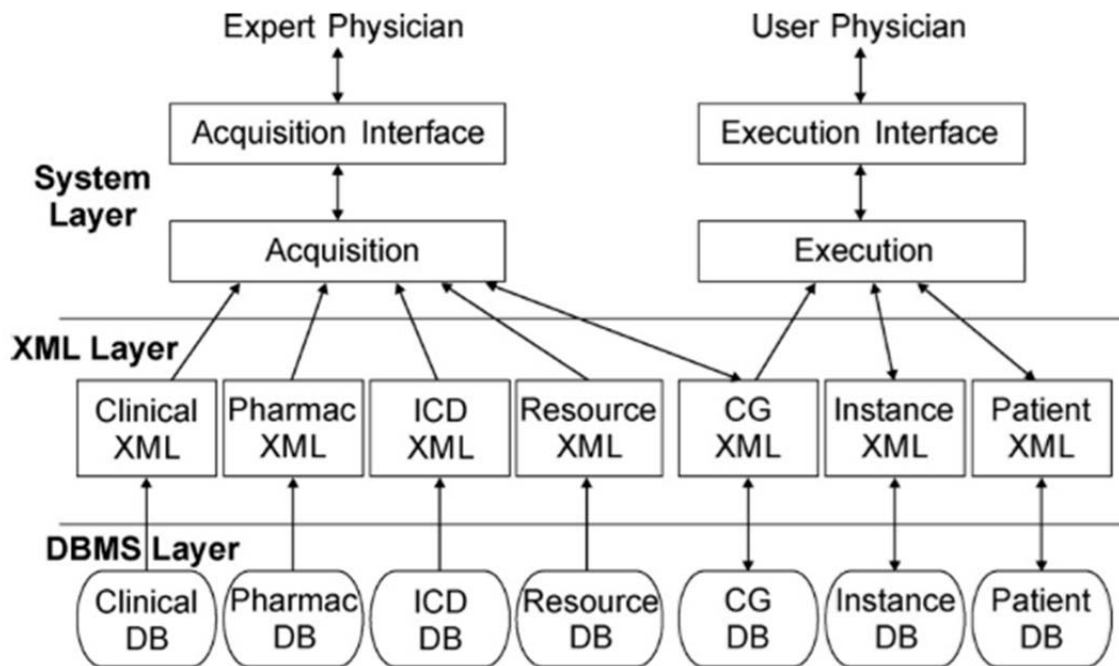


Abbildung 23 :Architektur von GLARE. Vierecke sind Berechnungsmodule und Kreise repräsentieren Wissensbasen (Datenbanken). Der linke Teil entspricht der Aquisition von Daten, der Rechte der Ausführinheit.

- CG-KRM : Clinical Guidelines Knowledge Representation Manager: Ist eine Wissenssammlung von internen Leitlinien und domänenunabhängig
- CG-DB : Clinical Guideline DataBase : Hier werden die Leitlinien des CG-KRM in Datenbanken gespeichert
- CG-AM: Clinical Guideline Aquisition Manager : Experten können hier über ein Benutzerfreundliches Interface Leitlinien im CG-KRM zu beschreiben
- CG-EM: Speichert den Status der Ausführungen eines klinischen Pfades in einer Datenbank und interagiert mit dem behandelnden Arzt mithilfe der CG-IM Nutzeroberfläche
- CG-IM : Bietet Ärzten ein Graphisches Interface für die Interaktionen mit dem CG-EM



Idee der oberen Architektur ist die möglichst unabhängige Ausführbarkeit und Unabhängigkeit von einem DBMS[23]. Die XML Zwischenschicht dient dabei als Interlingua zwischen DBMS und GLARE. DBMS und XML Layer verwalten die gleichen Daten, besitzen aber unterschiedliche Funktionalitäten. Dies ermöglicht ein verschmelzen der Vorteile beider Ansätze [101].

Sprachelemente

Die Sprache unterscheidet sogenannte **atomic** und **composite** Aktionen. Aktionen des Typs atomic müssen dabei unbedingt erledigt werden. Aktionen des Typs composite sind dabei Zusammensetzungen aus atomic Aktionen. Die Aktionen des Typs atomic werden dabei noch einmal in 4 Typen unterteilt. Diese kleine Anzahl von Primitiven soll die Verständlichkeit von klinischen Pfaden, die in GLARE codiert wurden, erhöhen.

Die 4 Typen von atomic Aktionen:

work action : Physische Aktionen mit der Umwelt die zu einem bestimmten Zeitpunkt der klinischen Pfade ausgeführt werden sollen. Diese Aktionen werden durch textuelle Attribute wie einer Beschreibung, den Kosten, der benötigten Zeit oder den zu verwendeten

Ressourcen beschrieben [76]. Ein Beispiel ist das Messen des Blutdruckes, die Verabreichung eines Medikamentes oder das Verabreichen einer Infusion.

query action : Ist eine Anforderungen von Informationen von der Außenwelt, z.B. von anderen Ärzten, Datenbanken und Wissensbasen. Beispiele hierfür sind das Anfordern von Laborproben oder das Anfordern der Anamnese des Patienten.

decision : Entscheidung , beruhend auf bestimmten Kriterien der Leitlinie . Es werden 2 Typen von decions unterschieden. Diese sind **diagnostic decicions** und **therapeutic decions**.

Diagnostic decisions:

Diagnostic decisions sind Entscheidungen, die mithilfe von Tripeln aus <Diagnose, Parameter Punktzahl> beschrieben werden, wobei ein Parameter wiederum aus Tripeln von <Data,Attribut,Value> besteht.

Zum Beispiel kann so bei einer Behandlung von Speiseröhren und Magenkrebs (Gastro Esophageal Reflux Disease kurz GERD) zwischen 3 verschiedene Möglichkeiten unterschieden werden. Diese sind „kein GERD“, „möglicherweise GERD“ oder „wahrscheinlich GERD“. Die Beispieltripel von <diagnose,parameter,value> können dann so beschrieben werden:

- <GERD,möglicherweise GERD,Punktzahl>
- <GERD,wahrscheinlich GERD,Punktzahl>
- <GERD,kein GERD,Punktzahl>

Diese Tripel werden dann zur Entscheidungsfindung nach einem punkteverfahren genutzt, wobei die Entscheidung mit der höchsten Punktzahl Priorität erhält. Die Punktzahlen werden durch die darunterliegenden Parametertripel ermittelt. Die Parameter sind einzelne gewichtete Symptome einer Krankheit.

Auf das Beispiel von GERD bezogen können so verschiedene Parameter, wie ein Gewichtsverlust oder das Vorhandensein von Sodbrennen oder Schluckbeschwerden, definiert werden.

Das Vorhandensein von Schluckbeschwerden und das Vorhandensein von Sodbrennen kann dann z.B. so als Parametertripel dargestellt werden:

- < wahrscheinlich GERD, Schluckbeschwerden vorhanden ,5>
- < wahrscheinlich GERD, Sodbrennen vorhanden, 10>

Oder

- < kein GERD, Schluckbeschwerden vorhanden ,0>
- < kein GERD, Sodbrennen vorhanden, 0>

Für jede Diagnosemöglichkeit muss dabei die Punktzahl definiert werden. Der letzte Wert des Tripels entspricht dabei der gewichteten Punktzahl des Parameters für eine bestimmte (erster Wert des Tripels) Diagnose und Symptom.

Ebenso können so Schwellenwerte definiert werden, ab denen eine Entscheidung in Frage kommt oder ausgeschlossen werden kann. Beispielhaft kann GERD bei einer Punktzahl von weniger als 10 Punkten ausgeschlossen werden, welches weitere Untersuchungen wie eine Endoskopie nicht mehr notwendig macht.

Therapeutic decisions: Die zweite Art sind therapeutic decisions. Diese nutzen fest vordefinierte Parameter wie Effektivität, Kosten, Nebenwirkungen, Dauer und Übereinstimmungen um Entscheidungen zu treffen. Beispiel für eine therapeutic decision ist z.B. die Behandlung von Gallensteinen bei denen bisher keine Krankheitssymptome aufgetreten sind. Diese können entweder operativ entfernt oder mit einem Medikament behandelt werden. Die Entscheidung bleibt dem Arzt oder auch nach Möglichkeit dem Patienten überlassen.

Diese Gruppe von Aktionen ermöglicht somit die Aufspaltung in Behandlungsalternativen von Leitlinien nach vorgegebenen Kriterien.

conclusion : Sie repräsentieren das Ergebnis von decisions . Sie folgen einer decision und stellen die einzelnen Entscheidungsmöglichkeiten dar. Somit können komplexere Entscheidungen getroffen werden, die nicht nur ja/nein Entscheidungen umfassen, wie z.B. die Annahme eines bestimmten Diagnosewertes bei einer vorherigen Untersuchung.

composite actions :

Composite actions sind zusammengesetzte Aktionen, die aus einer bestimmten Anzahl von atomic oder weiteren composite Aktionen bestehen. In der tiefsten Ebene bestehen diese ausschließlich aus Aktionen des Typs atomic. Dies soll die Übersichtlichkeit und Verständlichkeit der klinischen Pfade verbessern.

GLARE nutzt zur Darstellung von composite Aktionen einen Baum von Graphen:

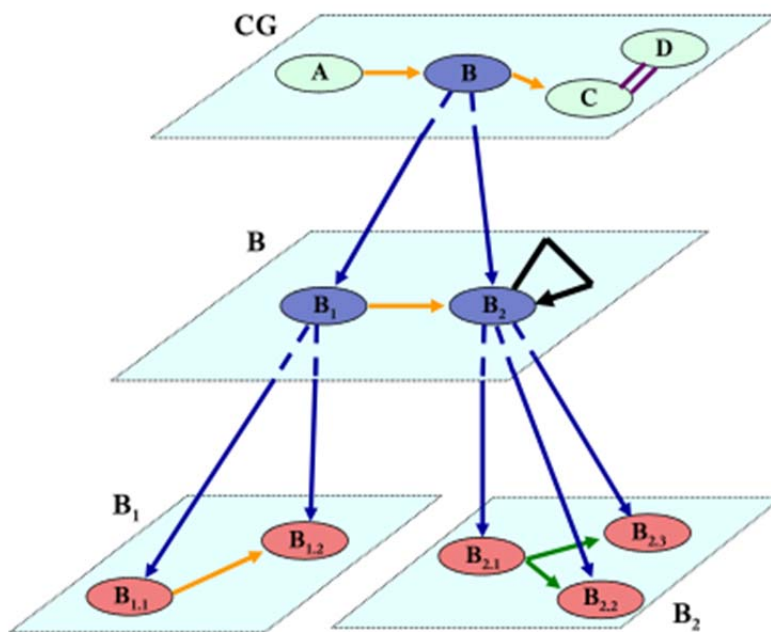


Abbildung 24 : GLARE Repräsentationsformalismus einer composite action

Eine beispielhafte Darstellung eines klinischen Pfades. Blau markierte Elemente stellen composite Aktionen dar, rote atomic Aktionen. A, C, D sind nicht genauer spezifiziert und dienen hier nur der Darstellung der Kontrollflussmöglichkeiten innerhalb der Leitlinie.

Für den Kontrollfluss existieren vier Relationen :

- orange : Sequenzieller Ablauf
- lila : Kontrolliert synchrone Ausführung (bsp. Während....) (C und D werden im Beispiel gleichzeitig ausgeführt)
- grün : alternativer Fluss (decision)
- schwarz : Wiederholung (Bsp: verabreichen von 3 Einheiten eines Medikamentes für den nächsten Monat)

Werkzeugunterstützung

Für die Erstellung der Leitlinien existiert ein Erfassungsmodul (CG-AM) welches nachfolgend Dargestellt ist. Für GLARE existieren keine frei zugänglichen Entwicklungswerkzeuge. Eine symbolische/sichtorientierte Erstellung der Prozesse über Programme wie LibreOffice oder Microsoft Visio kann nicht erfolgen, da darunterliegende Informationen nicht modelliert werden können.

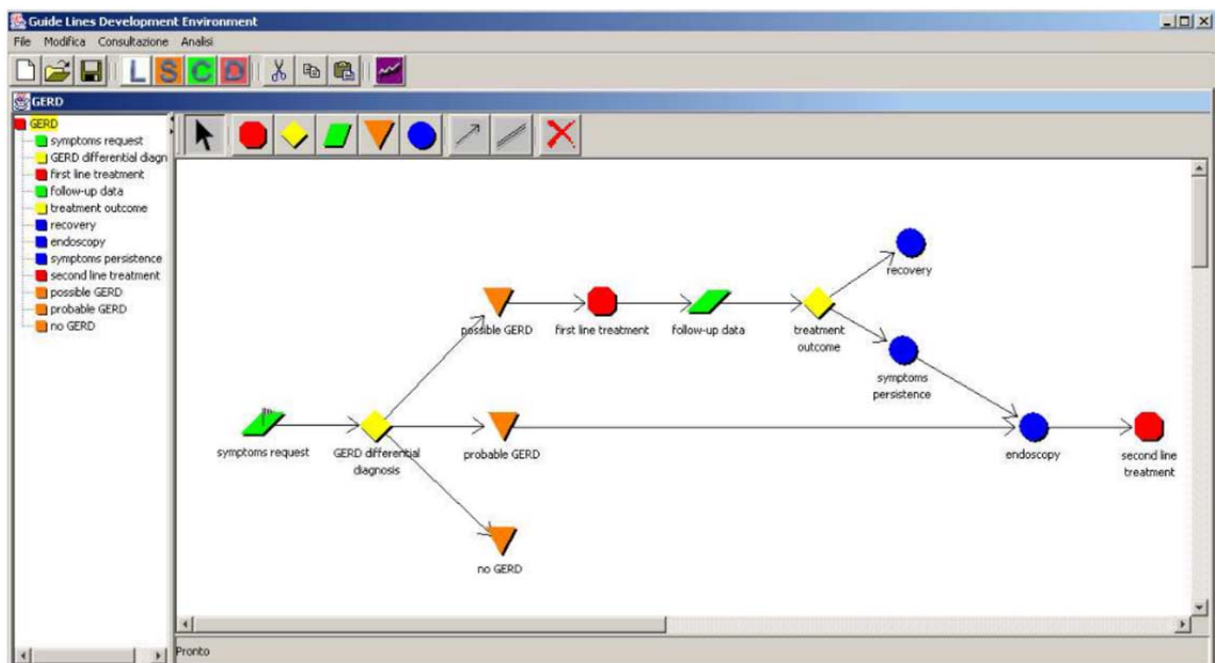


Abbildung 25 : Erfassungsmodul CG-AM. Gelbe Rauten stellen Decisions dar, Orange Dreiecke sind conclusions (Ergebnisse einer diagnostic decision), rote kreise sind composite Aktionen, blaue Kreise sind conclusions (Ergebnisse einer therapeutic DEcision) und grüne Vierecke sind Querys

Aktuelle Entwicklungen

Das GLARE Projekt war eine Kooperation zwischen dem Institut für Informatik und dem Krankenhaus von St. Johannes der Täufer in Turin. Auf den Webseiten beider Institutionen lassen sich jedoch nur noch sehr wenige Informationen zu GLARE finden. Das Projekt führte

zur Entwicklung eines Prototypsystems, welches unabhängig von der Domäne, an den Arzt ausgerichtet ist. Medizinischen Aufzeichnungen werden in diesem System auch einbezogen [75]. Das System wurde erfolgreich in verschiedenen medizinischen Behandlungen wie Blutkrebs, Herzversagen, Reflux getestet [72]. Das System nutzt laut Website inzwischen fortschrittliche Methoden der künstlichen Intelligenz, mit besonderem Schwerpunkt auf Entscheidungsunterstützung und Formen der modellbasierten Argumentation und zeitlichen Denkens [76]. Weitere Informationen zu diesem Prototypen werden nicht bereitgestellt (26.05.2010 Webaufttritt).

GLARE selbst diene dabei wohl der Grundlagenforschung und wird inzwischen nicht mehr weiterentwickelt. Die letzten Aktualisierungen fanden vor mehr als 5 Jahren statt.

3.4.6 GEM – Guideline Elements Model

Historie

GEM wurde 2000 im „Yale Center for Medical Informatics“ in der Yale University School of Medicine entwickelt. Mit GEM II wurde 2008 daraus ein ASTM Standard (E2210) geschaffen der die Codierung von Leitlinien als Dokumentenmodell in einem XML Datenformat beschreibt. Neuste Version ist GEM III, welches sich derzeit aber noch in der Entwicklung befindet.

Idee/Ziel

Idee von GEM ist es, ein Dokumentenmodell für die klinische Domäne bereitzustellen in der Leitlinien aus natürlicher Sprache semistrukturiert in das XML Datenformat umgewandelt werden können. Das Dokumentenmodell beschreibt dabei den Aufbau des Dokumentes, in den die Leitlinie überführt und gespeichert wird [87].

Ziel von GEM ist es den Spalt zwischen der Leitliniendefinition und implementierungsnahen Sprachen wie Arden zu schließen. Dazu werden XML Basierte Vorlagen von Attributen mit entsprechenden in den Leitlinien vorhandenen Informationen gefüllt. Es ist keine direkt Ausführbare Modellierungssprache, bietet aber mithilfe von Tools (GEM-Arden) eine Teilweise Umwandlung in die Arden Syntax.

GEM Modell

GEM besteht aus einer Hierarchie von Leitlinienkonzepten mit mehr als 100 Elementen von denen aus Platzgründen nur die wichtigsten Elemente vorgestellt werden. Diese Konzepte wurden international standardisiert um bei der Überführung von Leitlinien in das GEM Format möglichst wenige Interpretationsmöglichkeiten zu lassen.

Guideline Elements Model

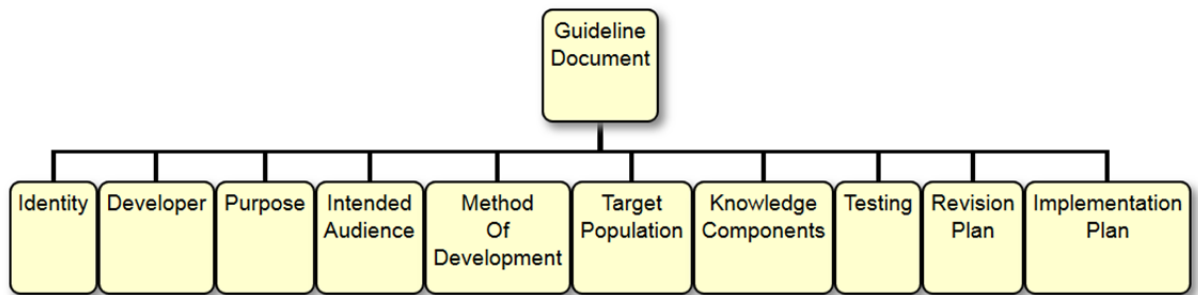


Abbildung 26 : GEM 3 Dokumentenarchitektur [36]

Das obere Beispiel zeigt das Wurzelement einer GEM Leitlinie in der zuerst viele organisatorische Aspekte in natürlichem Text gespeichert werden. Wichtigstes Attribut zur Codierung der Leitlinien ist das Knowledge Components Attribut welches nachfolgend genauer beschrieben wird.

Sprachelemente

GEM umfasst eine Vielzahl von Attributen [35] zur Codierung von Leitlinien in das XML Format. GEM Leitlinien umfassen dabei nicht nur Leitlinienwissen, sondern auch allgemeine Informationen wie Autor, Modifizierungsdatum und eine Modifizierungshistorie.

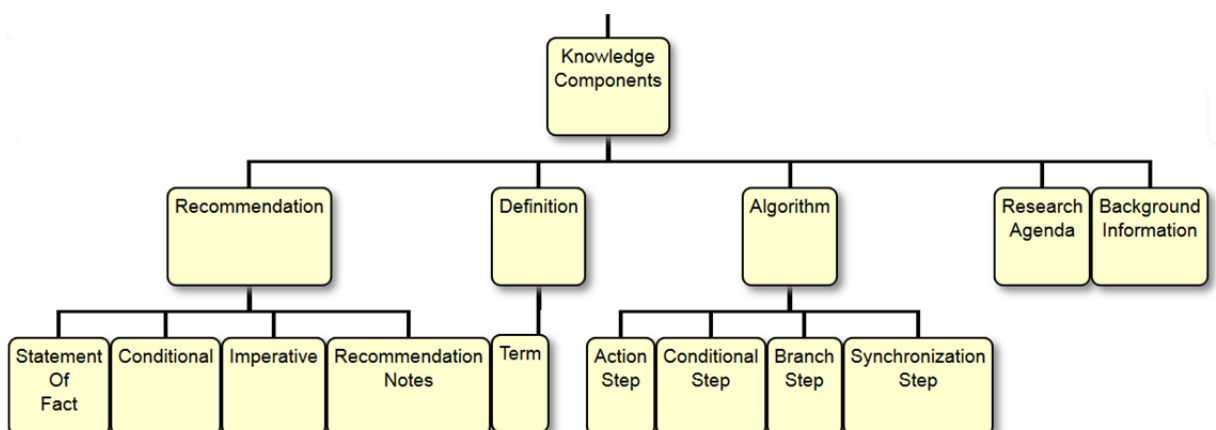


Abbildung 27 : GEM 3 Knowledge Komponenten

Für die Codierung der Leitlinien sind dabei die sogenannten Knowledge Components zuständig. Darin sind Elemente zur Ablaufsteuerung und Entscheidungsfindung untergebracht. Die fünf Kinderattribute umfassen:

- **Recommendation** : Empfehlungen für den Arzt

- **Terms** (Dient der Definition von Abkürzungen und verwendeten Ausdrücken)
- **Algorithm** : Dieses Attribut und dessen Kinder sind zuständig für die Ablaufsteuerung der Leitlinie analog zu GLIF
 - o Action Step : ist ein Einzelschritt der maximal einen Vor- bzw. Nachfolger besitzt
 - o Conditional Step : Entscheidungsschritt mit einem Vorgänger und mehreren möglichen Nachfolgern der mithilfe der „Criterion“ Eigenschaft evaluiert wird.
 - o Branch Step : Dient der Aufspaltung des Kontrollflusses der Leitlinie in mehrere unabhängige Kontrollflüsse
 - o Synchronization Step : Dient der Vereinigung von Kontrollflüssen die über einen Branch Step aufgeteilt wurden.
- **Research Agenda** und **Background Information**: sind Hintergrundinformationen, weiterführendes Material, Beschreibungen und Links zu Quellen in normalen Text.

Beispiel (Frei erfunden): Da die meisten Beispiele wesentlich komplexere Entscheidungssituationen umfassen wird hier ein frei erfundenes Beispiel mit einer minimalen Anzahl von Elementen vorgestellt.

```

<Recommendation> Empfehlung 1
  <Conditional> Personen mit Bluthochdruck wird von der AMWF die Verwendung eines
  Blutdrucksenkenden Mittels wie MittelXYZ empfohlen.
    <Decision.variable>Bluthochdruck</Decision.variable>
    <Action> Verabreichung von Mittel XYZ</Action>
    <Reason> wird von der AMWF empfohlen</Reason>
  </Conditional>
</Recommendation>

```

Recommendation erhält den Namen der Empfehlung hier „Empfehlung 1“ und ist frei wählbar. Danach wird über das **Conditional** Attribut eine Entscheidungsfindung eröffnet und der Leitlinientext als Freitext gespeichert. Dann wird nach dem Attribut **Decision.variable** evaluiert. Liefert die Evaluierung ein positives Ergebnis soll als **Action** das Mittel XYZ verabreicht werden. Dies wird durch das **Reason** Attribut unterstützt, welches weiterführende Informationen wie die Empfehlung von der AMWF dazu enthält.

Werkzeugunterstützung

Das Werkzeug zum Arbeiten ist GEM-Cutter, ein auf Java basiertes Programm, entwickelt und gepflegt vom Yale Center of Medical Informatics at Yale University School of Medicine.

Dieses Programm ist nach einer Registrierung erhältlich. Das Tool ist an die aktuelle Version 3 von GEM angepasst und nutzt das Java Runtime Environment 1.7. Ebenso wird der GEM-Transformer bereitgestellt um GEM 1 Leitlinien einfach auf GEM 2/3 zu aktualisieren. [37]

Aktuelle Entwicklungen

GEM wurde zu einem ASTM (E2210 [12]) Standard zur Repräsentation von Leitlinien im XML Format [96]. Das GEM Modell selbst wird weiterhin vom Yale Center of Medical Informatics at Yale University School of Medicine gepflegt und befindet sich derzeit in Version 3. Die Website National Guideline Clearinghouse [67] erstellt dazu gerade GEM-Cut Versionen von ausgewählten Leitlinien [27], die aber zum Zeitpunkt der Arbeitserstellung noch nicht zur Verfügung standen.

Mitglieder des GEM Projektteams, die am Yale Center for Medical Informatics lehren, partizipieren ebenso bei den Bemühungen von HL7 zur Leitlinienmodellierung. [96]

3.4.7 GLIF - Guideline Interchange Format

Historie

GLIF entstand als Gemeinschaftsprojekt des InterMed Collaboraty („collaboration“ + „laboratory“) bestehend aus der Harvard University Abteilung Medical Informatics, der Columbia University und der MCGill University in den USA. Die erste öffentliche Version wurde mit GLIF 2 im Jahr 1998 publiziert. Derzeit ist die neuste Version GLIF 3.5 die im Mai 2004 veröffentlicht wurde. Ursprünglich basiert die Sprache auf dem Arden Syntax, der mit GLIF 3 in einen eigenen an Arden angelehnten Ansatz umgewandelt wurde [16].

Idee/Ziel

Idee hinter der Entwicklung von GLIF war die Entwicklung eines standardisierten und austauschbaren Formates, da die Leitlinienentwicklung mit sehr viel Kosten verbunden ist, die so gesenkt werden könnten. Leitlinien können so als allgemeine Basis für lokale Adaptionen genutzt werden und zum Austausch von Leitlinien unabhängig von Plattform und System genutzt werden [58].

Das Ziel von GLIF ist die Bereitstellung von Leitlinien, die in einem computerinterpretierbaren Format gespeichert werden und über verschiedene klinische Bereiche, wie Behandlung, Management und Verwaltung genutzt werden können. Für den Austausch und die Speicherung von Leitlinien wird das XML Datenformat verwendet. In GLIF werden 3 Abstraktionsebenen, Level A, B und C, unterschieden ((A)Konzeptebene, (B)Maschinenebene, (C)Medical Knowledge Layer), die im folgenden Abschnitt genauer beschrieben werden [65]. Dabei baut jede der höheren Ebenen auf die Konstrukte der darunterliegenden Ebenen auf.

Die 3 Abstraktionsebenen von GLIF

Level A - Konzeptebene

Leitlinien werden visuell als Flussdiagramme dargestellt. Diese Diagramme können zum durchstöbern und zur Navigation innerhalb der Leitlinien verwendet werden. Diese Abstraktionsebene kann aber nicht für die computergestützte Ausführung von Leitlinien, noch für Entscheidungshilfssysteme, verwendet werden. Eine Graphische Darstellung erfolgt

nicht standardisiert, da für die einzelnen Elemente keine vorgegebenen Symboliken existieren, wie die nachfolgenden Beispiele zeigen.

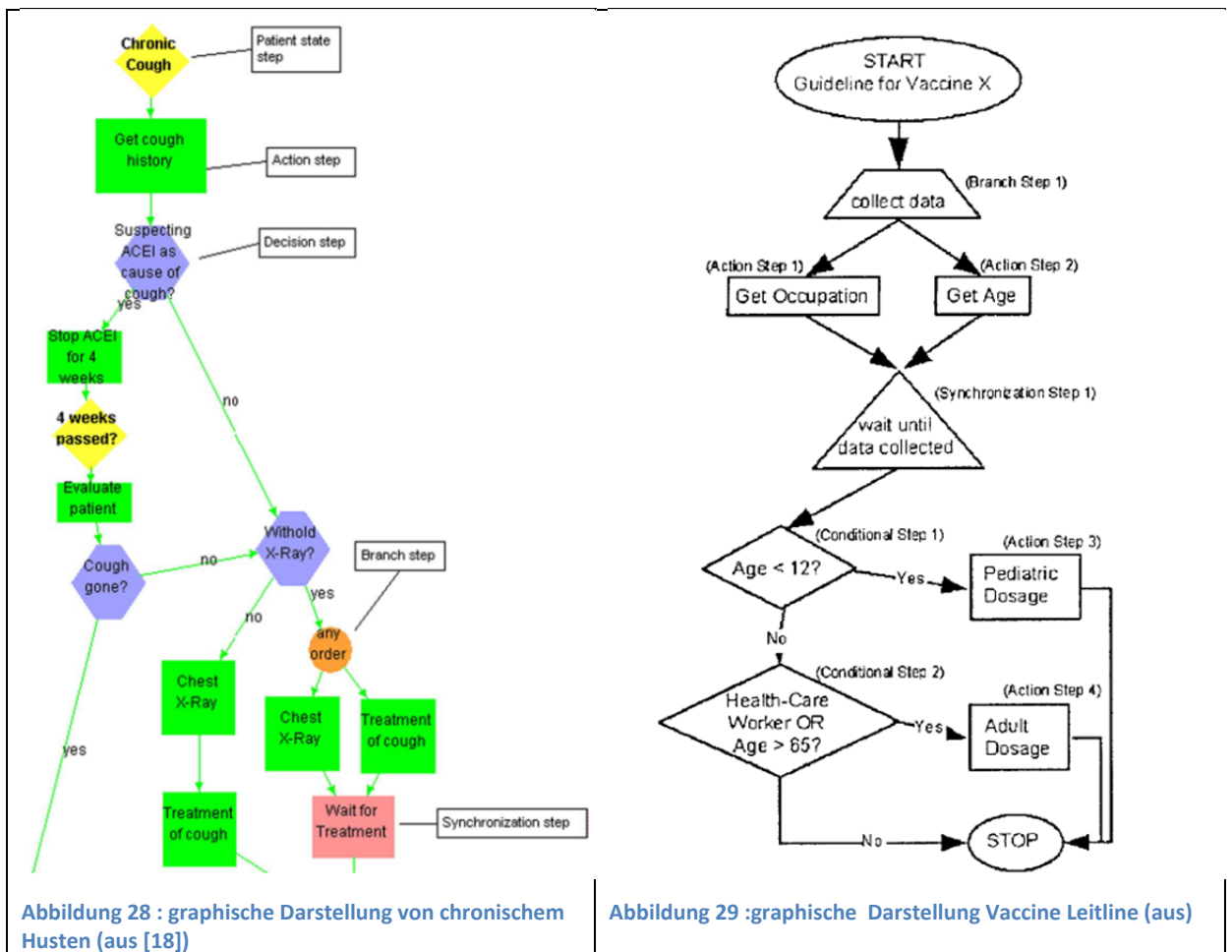


Tabelle 3 : Vergleich graphischer Darstellungen von GLIF 3

Obige Darstellungen zeigen verschiedene Möglichkeiten der visuellen Darstellung von Leitlinien innerhalb der GLIF konzeptebene.

Level B - Maschinenebene

Leitlinien werden als berechenbares Modell aus Ausdrücken, Klinischen Handlungen und Leitlinienkontrollflüssen dargestellt. Somit wird eine logische Prüfung auf Konsistenz und Komplettheit ermöglicht. Hier findet die eigentliche Definition von Patientendaten, klinischen Aktionen und Abläufen statt. Diese ebene nutzt das Reference Information Model (RIM) von HL7 um den Austausch und die Ausführung von Leitlinien zu ermöglichen und nutzt das im Level-C definierte Wissen zur Algorithmierung der Leitlinie.

„Nesting“ erlaubt nun auch seit Version 3 von GLIF eine Einbettung/Gruppierung ganzer Behandlungen in andere Behandlungen. Somit werden verschiedene Granularitätsstufen eingeführt die zu einer verbesserten Lesbarkeit von komplexen Leitlinien führen soll. Ebenso können Gruppierte Teilabschnitte leichter wiederverwendet werden.

Bei der Planung und Erstellung von Leitlinien können so z.B. auch allgemeine Aktionen durch detaillierte Prozeduren ersetzt werden.

Sprachelemente

Die GLIF 3 Spezifikation definiert ein objektorientiertes Modell, welches eine Anzahl von Klassen zur Beschreibung von Leitlinien nutzt und einem GLIF Syntax der den Aufbau und die Grammatik von Sprachkonstrukten definiert.

GLIF 3 Modell

Das GLIF 3 Modell besteht aus einer Reihe von Klassen für Entitäten der Leitlinie, deren Attributen und den damit verbundenen Datentypen. Jede in GLIF 3 abgebildete Leitlinie ist dabei eine Instanz des Allgemeinen Leitlinienmodells (Guideline Model). Nachfolgend werden die wichtigsten Klassen des GLIF 3 Modells aus der Vogelperspektive dargestellt und erläutert.

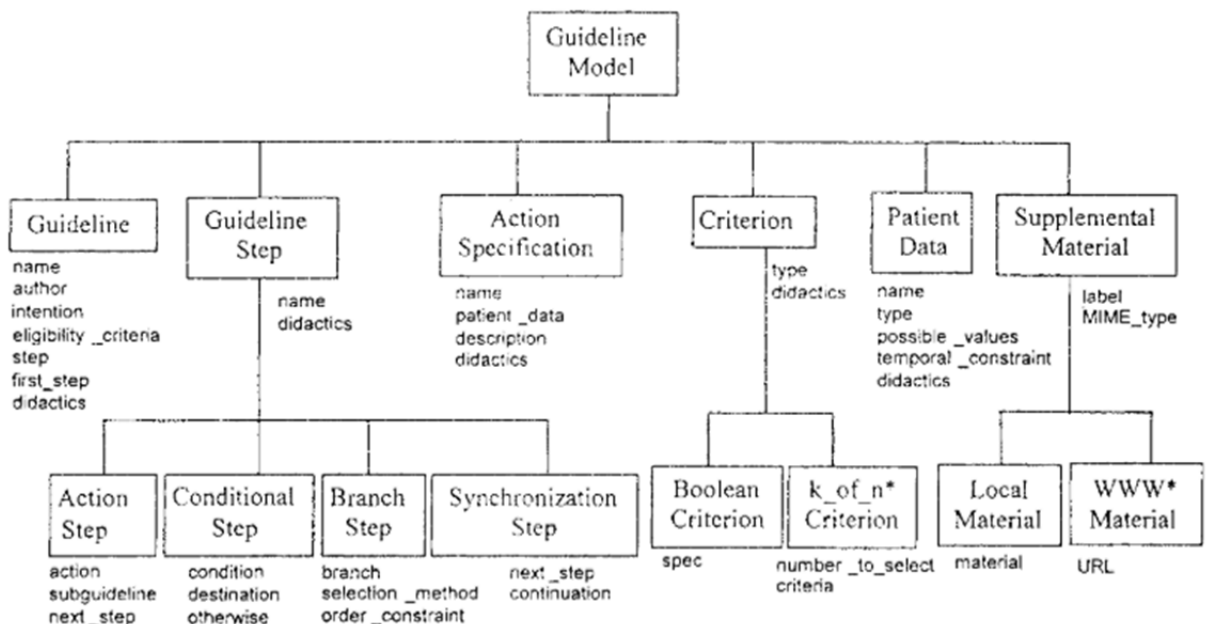


Tabelle 4 : TOP Level Ansicht der GLIF Klassen und Attribute

Erläuterungen

Guideline : Eine Leitlinie besitzt allgemeine Attribute wie ihren Name(name), den Autor(author), das Ziel oder Verwendungszweck der Leitlinie(intention), meistens als normaler Text, Voraussetzungen an den Patienten(eligibility_criteria) als eine Liste von Criterion Objekten sowie didaktische Informationen (didactics) und den ersten Behandlungsschritt (first_step).

Guideline Step: Eine Leitlinie besteht aus einer Anzahl von Aktionen die in einem direkten Graph abgebildet werden können. Ein Guideline Step kann dabei aus 4 verschiedenen Typen von Schritten bestehen :

- **Action Step** : Beschreibt klinische Aktionen zur Patientenbehandlung. Jeder Schritt enthält genau eine Anweisung und einen Zeiger auf den nächsten Schritt. Ebenso können über Action-Steps auf Subguidelines verwiesen werden, um einen höheren detailgrad der Aktion zu erhalten.
- **Conditional Step** : Entscheidungen innerhalb einer Leitlinie werden mit Conditional Steps modelliert. Sie verbinden über Bedingungen (criterion) einen Schritt einer Leitlinie mit einem ausgewählten anderen Schritt. Das „criterion“ ist eine logische Aussage die ein true/false Ergebnis liefert. Wird die Aussage des „criterion“ erfüllt so wird der Schritt unter dem „destination“ Attribut ausgeführt. Ergibt er false wird der Schritt des „otherwise“ Attributes ausgeführt. Wenn keine ausreichenden Daten für die Entscheidung vorhanden sind wird keiner der beiden Schritte ausgeführt.
- **Branch Step** : Dieser Typ von Aktion dient der Parallelisierung und Aufspaltung des Kontrollflusses von verschiedenen gleichzeitig ausführbaren Behandlungsschritten. Sie bilden Verbindungen zu mehreren Leitlinienschritten, in denen der Autor spezifiziert ob alle Schritte, einige oder nur einer der Schritte ausgeführt werden soll. Ebenso kann der Autor bestimmen ob diese Schritte in einer bestimmten Reihenfolge oder parallel ausgeführt werden können.
- **Synchronization Step** : Werden in Verbindung mit „Branching Steps“ genutzt. Voraussetzung für einen „Synchronisation Step“ ist, dass ein Branching Step den Kontrollfluss in mehrere einzelne Kontrollflüsse aufteilt. Diese können dann mithilfe dieses Schrittes wieder in einen Kontrollfluss zusammengeführt werden. Innerhalb dieses Schrittes kann mithilfe des „continuation“ Attributes bestimmt werden, ob alle

oder nur ein paar oder einer der Schritte am synchronization step angekommen sein müssen.

Action Specification: Enthält den Name einer Aktion sowie eine Beschreibung der Aktion in normalem Text und eine Liste von benötigten Patientendaten. Optional sind auch Links zu weiterführendem didaktischen Material im „didactics“ Attribut speicherbar.

Criterion : Dient der Entscheidungsfindung. Dieses Klasse beschreibt Bedingungen, die true oder false ergeben können und die erfüllt sein müssen, bevor z.B. ein Patient eine Behandlung nach einer bestimmten Leitlinie erfährt.

Patient Data: Wenn eine Aktion das Erfassen von Patientendaten erfordert, so wird ein „Patient Data Element“ definiert, welches einen Namen und einen Datentyp besitzt um diese Informationen aufzunehmen. Optional kann eine Liste von akzeptierbaren Werten mitgegeben werden. Als Beispiel hierfür kann das Alter des Patienten herangezogen werden, welches einer Zahl größer 0 entsprechen muss (alter>0) oder eine Temporale Bedingung wie das letzte vorhandene Messergebnis (temporal(letztes Messergebnis)).

Supplemental Material : Dieses Objekt speichert zusätzliche didaktische Materialien als normalen Text, Video, Bild oder in Form von URLs für weiterführende Informationen zu der Leitlinie.

GLIF 3 Syntax

Die GLIF 3 Syntax besitzt eine an die Arden angelehnte Grammatik . Diese Grammatik wird zur Repräsentation von Entscheidungskriterien, auslösenden Ereignissen, medizinischem Wissen und klinischen Aktionen verwendet. Idee dabei ist die Definition von abstrakten medizinischen Wissen um nicht von bestimmten Informationssystemen abhängig zu sein. Dies soll auch den Austausch von Leitlinien über verschiedene Institutionen ermöglichen. Die Spezifikation für ergänzendes Material erlaubt es Lehrmaterial mit der Richtlinie zu verbinden und einem bestimmten Abschnitt der Leitlinie zuzuordnen. Dieses Zusatzmaterial kann in verschiedenen Formaten wie z.B. HTML oder normalen Text gespeichert und für verschiedene zwecke wie Grundlagen, weiterführende Literatur oder für die Patientenaufklärung verwendet werden.

Kurzes Beispiel der GLIF 3 Syntax : „Erwachsener Junger Nichtraucher oder schwangere Frau“

((Age > 18 year) and smokingEndTime < now)) or (pregnancyDueDate > now)

Vom Arden Syntax verwendete Konstrukte sind in blau geschrieben. Grüne Elemente beschreiben die in Level – C definierten medizinischen Konzepte.

Level C - Medical Knowledge Layer

(in Entwicklung/nicht fertiggestellt): Hier erfolgt die Abbildung von Leitlinien auf Elektronische Patientenakten und Prozeduren der Informationssysteme einer spezifischen Institution sowie die Definition von medizinischen Konzepten, die von den darüber liegenden Ebenen genutzt werden. Diese Ebene enthält auch nicht austauschbare Elemente die auf das jeweilige Institut angepasst sind.

Diese Ebene beschreibt Methoden, wie das medizinische Wissen an Wissensquellen angebunden werden und genutzt werden soll. Da sich diese Ebene in der Entwicklung befindet existieren dazu vorerst nur 4 konzeptuelle Regeln für die Erstellung von Konzepten:

- 1. Das Vokabular, welches medizinische Konzepte textuell definiert und diese mit eindeutigen identifikationsmerkmalen versieht (Beispiel, Definition von Alter, Gewicht, Größe einer Person), die für spätere Entscheidungsfindungen genutzt werden können
- 2. Wissensbasen welche medizinisches Wissen wie Medikamentenhierarchien und zulässige Intervalle für Testresultate definiert
- 3. Die Anbindung sowie das Mapping an vorhandene Wissensbasen wie z.B. die elektronische Patientenakte (EMR)
- 4. Andere klinische Anwendungen wie Auftragserfassung sowie Alarm und Erinnerungssysteme

Eine „standalone“ Ausführung ist bei GLIF kein erstrangiges Ziel. Vielmehr sollen Module entwickelt werden, die in bestehende Krankenhausinformationssysteme integriert werden und somit die Therapieplanung unterstützen.

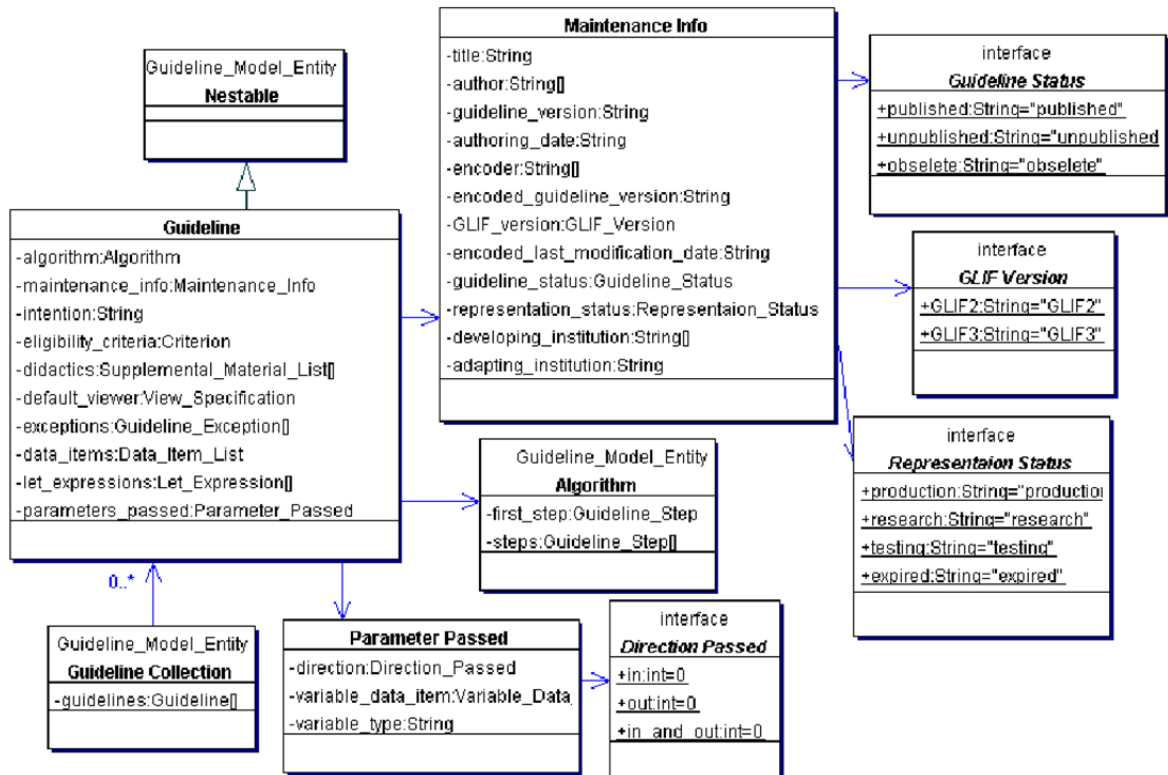


Abbildung 30 : Beispielhafte Darstellung für den Medical Knowledge Layer aus der GLIF 3.5 specification Manual in UML

Werkzeugunterstützung

GLIF nutzt das Werkzeug Protegé zur Entwicklung seiner Leitlinien. Dabei wird von der Specification Manual empfohlen das RIM von HL7 und dessen Klassen für die Konstrukte des Level C zu nutzen. Dabei ist zu beachten, dass das RIM immer noch Iterationen zur Verbesserung des Modells durchläuft und bei der Entwicklung soll deswegen vorher eine bestimmte Version des Modells herausgesucht werden um Kompatibilitätsprobleme zu vermeiden.

GLEE ist ein Werkzeug zum Ausführen von Leitlinien die in GLIF 3 codiert wurden. Dieses Tool steht nicht mehr zum Download zur Verfügung, daher können keine weiteren Informationen dazu gegeben werden.

Aktuelle Entwicklungen

Die Offizielle Website von GLIF wurde von einem Unternehmen übernommen, welches nun HCG Diäten anbietet und keine Informationen zu GLIF mehr liefert. Auf keiner der offiziellen

Seiten der beteiligten Universitäten lassen sich Hinweise auf GLIF finden. Lediglich auf der Website der Universität Hava in Israel lassen sich die Spezifikation Manuals herunterladen [66], wobei der Webaufruf seit 2010 nicht mehr aktualisiert wurde. Publikationen die sich speziell mit GLIF befassen gibt es seit 2004 nicht mehr und weitere Publikationen die auf GLIF verweisen, erwähnen diese Sprache nur am Rande. Ebenso sind viele der Publikationen nur noch als Titelangabe auf den Webseiten der Institutionen sowie in Zitationssuchmaschinen verfügbar.

3.4.8 (New)GUIDE

Historie

GUIDE wurde 1998 an der University of Pavia , Italien und der Ministero dell Istruzione dell Universita'e della Ricerca entworfen. GUIDE wurde 4 Jahre später als NewGUIDE in 2002 neu entwickelt. Im folgenden Kapitel wird NewGUIDE beschrieben.

Idee/Ziel

Idee von NewGUIDE ist die Schaffung eines Komplettsystems für Ausbildungs- sowie Erklärungszwecke für Personal der medizinischen Domäne.

Ziel von NewGUIDE ist, ein standalone Werkzeug bereitzustellen, welches das Personal der medizinischen Domäne auf spätere Workflow Managementsysteme und Krankenhausinformationssysteme vorbereitet, insbesondere in Verbindung mit der elektronischen Patientenakte [85].

NewGUIDE wurde als Kompletmanagementumgebung für computerinterpretierbare Leitlinien konzipiert [74]. Von Zertifizierten Gesundheitsorganisation wie z.B. dem National Guideline Clearing House werden dann Leitlinien bereitgestellt, damit die Qualität der zur Verfügung gestellten Leitlinien gewährleistet ist.

Sprachelemente

GUIDE orientiert sich als Modellierungssprache an Flussdiagrammen, erweitert diese aber noch um weitere Konstrukte um spezielle medizinische Sachverhalte abzubilden. Diese umfassen z.B. subjektive Entscheidungen, die von dem Behandelnden Arzt subjektiv entschieden werden müssen. Ebenso wird durch eine Unterscheidung in atomare, also einzelne Behandlungsschritte und zusammengesetzte Prozesse eine Strukturierung komplexerer Prozesse ermöglicht. GUIDE unterstützt die Integration von Leitlinien in die Organisationsstruktur mittels Entscheidungsbäumen. Jedes Modell in GUIDE kann in Petri-Netze überführt werden und so simuliert und auf Plausibilität geprüft werden.

NewGUIDE nutzt die Formalismen des Guideline Elements Model (GEM) zur Strukturierung seiner Leitlinien.

NewGUIDE unterscheidet grundsätzlich 3 Abstraktionsebenen:

- **General Task Frame:** Betrachtung des Behandlungsprozesses aus einer allgemeingültigen Perspektive ohne organisations- oder patientenspezifische Attribute zu beschreiben.
- **Specific Task Frame :** Durch die Ergänzung und Belegung mit zusätzlichen Attributen entsteht der organisationspezifische klinische Pfad
- **Actual Task :** Instanz eines klinischen Pfades durch aktuelle Daten eines konkreten Patienten

GUIDE besitzt keine eigene Ausführbarkeit. Die Ausführbarkeit wird durch die Darstellung mittels Petri-Netzen und Arbeitsabläufen in Form von Flussdiagrammen impliziert [74].

Einige Werke verweisen auf das NewGUIDE Tool, welches eine graphische Oberfläche zum Erstellen und Bearbeiten von Leitlinienflussdiagrammen ermöglicht. Es nutzt Flussdiagramme, um eine „gute theoretische Basis“ [22] für das Management von Prozessen zur Verfügung zu stellen.

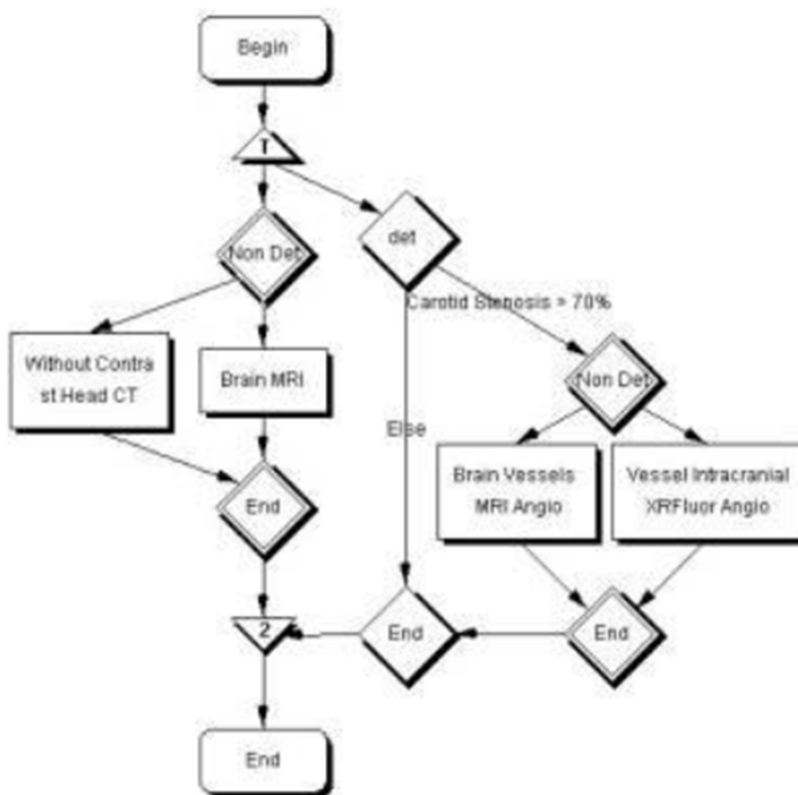


Abbildung 31 : Ausschnitt einer im NewGUIDE Tool Erstellten Leitlinie (Diagnosevorgang)

Intern werden Leitlinien sowie Pfade durch 9 relationale Tabellen einer Datenbank repräsentiert die hier nicht genauer beschrieben werden können [97]. Diese Tabellen umfassen alle notwendigen Elemente zur Codierung einer Leitlinie in einem relationalen Format.

Werkzeugunterstützung

NewGUIDE ist ein Komplettsystem welches in Java 2 Enterprise Edition (J2EE) implementiert wurde und mithilfe des Simple Object Access Protocol (SOAP) mit vorhandenen Krankenhausinformationssystem kommuniziert. Das NewGUIDE Tool ist nicht öffentlich zugänglich und kann deswegen nicht genauer beschrieben werden. Diese Tool dient der umfassenden Erstellung von Leitlinien in der GUIDE Modellierungssprache

Aktuelle Entwicklungen

NewGUIDE wurde prototypisch in 3 Anwendungen implementiert.

- Dekubitusprophylaxe
- Behandlung des akuten ischämischen Schlaganfalls
- Herzinsuffizienz-Management

Darüber hinaus werden in den Publikationen keine weiterführenden Verwendungszwecke beschrieben [74].

Die Webseiten der beteiligten Organisationen beschreiben NewGUIDE nur allgemein und bieten keine weiterführenden Informationen sowie Publikationen zu diesem System an.

3.4.9 Prestige

Historie

Prestige war ein Forschungsprojekt von 1996-1999. Prestige wurde dabei vom „EU 4th Framework Health Telematics Programme“ unterstützt. Beteiligt waren weltweit über 30 kommerzielle sowie nicht kommerzielle Organisationen aus 8 Ländern, welche für die Entwicklung von Leitlinien zuständig sind. Unter anderem war das deutsche Herzzentrum der Mainzer Universität beteiligt.

Idee/Ziel

Idee von Prestige war die Entwicklung eines Konzeptuellen Modells für den Informationsaustausch von Krankenhausinformationssystemen, welches weltweit durchgeführt wurde, um medizinischen Personal bei der täglichen Arbeit und der Best Practice zu unterstützen [102].

Ziel von Prestige war mittels Informationssystemen dem Personal der medizinischen Domäne im täglichen Leben Hilfe bei der Behandlung von Patienten zu geben.

Prestige enthält eine Anzahl von generischen Leitlinien und Patientendaten sowie Enterprise Informationen welche unabhängig von Sprache und medizinischer Domäne definiert sind. Ein Behandlungspfad wird aus medizinischer Sicht beschrieben und kann später über das „Interchange Format“, welches jedoch nicht genauer beschrieben wird, ausgetauscht werden. Das Projekt wurde mit der Aussage geschlossen „es wäre ein hohes theoretisches Potential vorhanden“ [38], aber 2 Flaschenhälse würden sich ergeben . Diese waren :

- 1.)Eine Überführung in das von Prestige genutzte Format aller Leitlinien
- 2.)Kompatibilität zwischen den vorhandenen KIS und Prestige

Die Quellenlage zum Prestige Projekt ist sehr dürftig. Offizielle Seiten existieren nicht mehr. Bis auf wenige allgemeine Publikationen konnten keine weiteren Informationen gefunden werden. Aus diesem Grund kann die Sprache nicht mit in die Betrachtungen einbezogen werden. Andere spätere Publikationen erwähnen Prestige auch nur oberflächlich.

Desweiteren beschreibt Prestige, dass weitere Arbeiten zur Weiterentwicklung eines einheitlichen Modells notwendig sind um konzeptuelle, kulturelle und klinische unterschiede verallgemeinern zu können [38]. Ebenso wurden während des Projektes XML und RDF als Webstandards etabliert, welche von dem Projekt nicht beachtet werden konnten und auf deren Basis eine Vereinheitlichung der Modelle vereinfacht würde.

Sprachelemente

Prestige besitzt kein Modell welches einzeln zur Ausführung von Leitlinien geeignet ist. Prestige beschreibt die Elektronische Patientenakte und welche Elemente darin enthalten sein müssen, um sie in den damals bestehenden Informationssystemen einsetzen zu können.

Prestige war eines der ersten Forschungsprojekte, welche eine Integration mithilfe der Elektronischen Patientenakte in bestehende Krankenhausinformationssysteme unternahm und dafür ein einheitliches konzeptuelles Modell entwickelte. Dazu wurde ein minimales Modell der elektronischen Patientenakte entwickelt, welches den Anforderungen der medizinischen Domäne gerecht werden sollte.

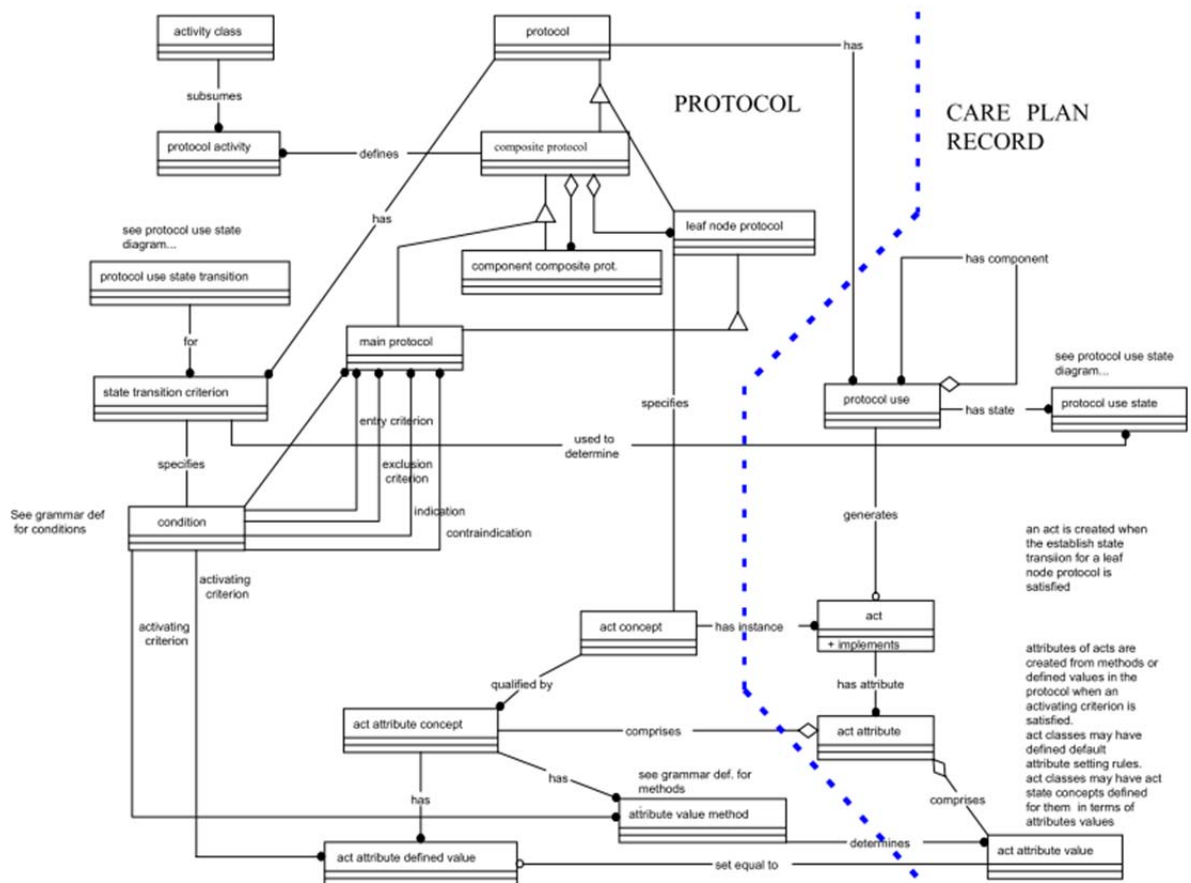


Abbildung 32 : Syntax und Semantik des Prestige Protocol Modells

Die obige Abbildung zeigt das von Prestige entwickelte Informationsmodell, das die grundlegenden notwendigen Elemente des elektronischen Protokolls zeigt. Dabei handelt es sich aber mehr um ein Klassendiagramm mit wenigen Sprachkonstrukten und deren Generalisierungsmöglichkeiten.

Werkzeugunterstützung

Das Tool GAUDI sowie GLEAM wurden bei der Entwicklung der von Prestige verwendeten Modelle genutzt. Beide Werkzeuge stehen nicht mehr zur Verfügung.

Aktuelle Entwicklungen

Da das Prestige Projekt im Jahre 1999 eingestellt wurde kann zu den derzeitigen Entwicklungen wenig Neues berichtet werden. Das Prestige Projekt wird in den meisten Publikationen nur am Rande erwähnt und dient als theoretische Basis für weiterentwickelte Informationsmodelle der medizinischen Domäne.

3.4.10 Prodigy

Historie

Prodigy wurde am „Sowerby Centre for Health Informatics at Newcastle (SCHIN)“, ein Teil der Newcastle Universität, mit Unterstützung des „Departement of Health (UK)“ in den Jahren 1996-1999 entwickelt. SCHIN wurde nach Beendigung des Projektes kommerzialisiert um weiteren Support leisten zu können. Prodigy selbst wird nicht mehr weiterentwickelt.

Idee/Ziel

Idee von Prodigy war die Entwicklung eines einfachen Modells, welches ausschließlich für chronischen Krankheiten gedacht ist.

Ziel von PRODIGY ist das Erleichtern der Wissensverarbeitung durch die Bereitstellung eines einfachen verständlichen Modells für einen speziellen medizinischen Teilbereich. Das Modell von Prodigy beschränkte sich ausschließlich auf chronische Krankheiten. Mehrere hundert Modelle wurden bereits im Prodigy Modell erstellt. Da Prodigy in kommerziellen Produkten eingesetzt wird, sind keine detaillierten Informationen zu den aktuellen Modellen sowie Ausführheiten verfügbar. Prodigy wird nicht weiterentwickelt und kann nicht oder nur teilweise mit in die Betrachtungen einbezogen werden, da keine ausreichenden aktuellen Informationen verfügbar sind[13].

Werkzeugunterstützung

Für Prodigy stehen keine frei zugänglichen Werkzeuge zur Verfügung.

Aktuelle Entwicklungen

Prodigy wurde als „NICE“ [70] neu belebt und wird von „NICE Evidence Services“ als WebPlattform zur Entscheidungshilfssystem auf Browserbasis angeboten [69]. Dieses System ist für jeden im englischen Gesundheits- und Sozialwesen gedacht, der medizinische Entscheidungen fällen soll. Das System wurde primär für Ärzte und Pfleger entwickelt, bietet aber auch Patienten oder nicht registrierten Personen Zugang zu einem Großteil des Inhaltes. Es werden inzwischen auch Leitlinien von nicht chronischen Erkrankungen bereitgestellt. Über Aufbau und Architektur des Systems lassen sich keine weiterführenden Informationen finden.

Beispiel von der NICE Website

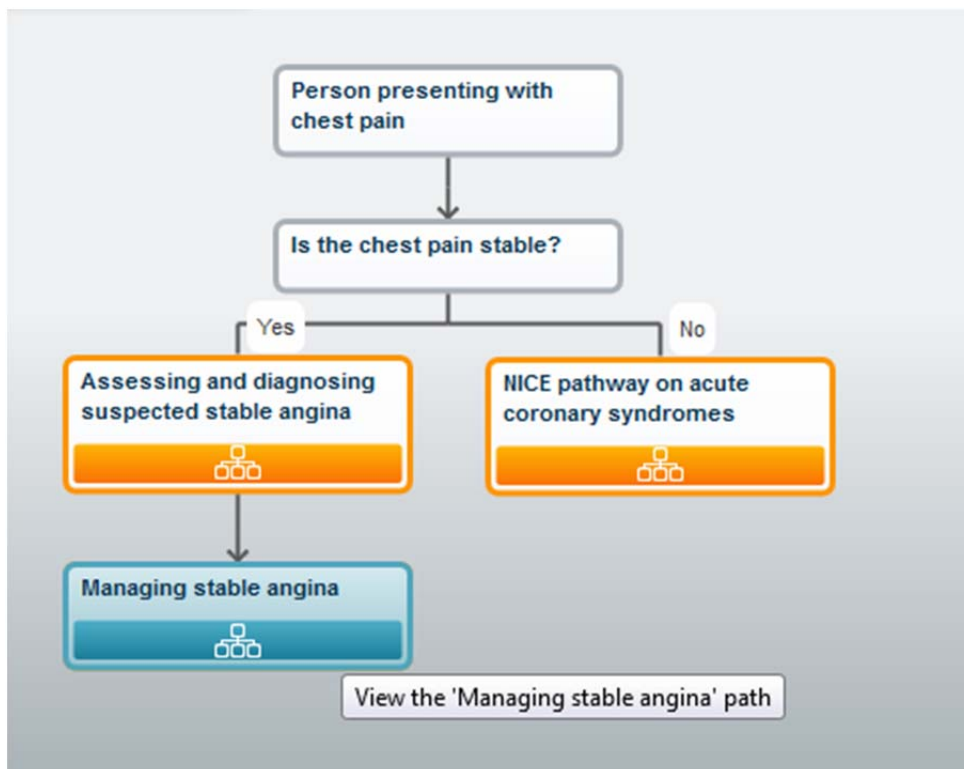


Abbildung 33 : NICE Plan für die Behandlung einer Angina

Leitlinien werden in NICE in einer an Flussdiagrammen orientierten Sicht dargestellt. Pläne mit den Netzwerksymbolen stellen weitere detailliertere Sichten einer Leitlinie zur Verfügung die den gleichen Aufbau verfolgen. Entscheidungen basieren auf Ja/Nein Entscheidungen.

3.4.11 PROforma

Historie

Die Entwicklung von PROforma startete im Jahr 1992 am „Advanced Computation Laboratory ‚Cancer Reaserch UK“. Nach ständiger Weiterentwicklung wurden im Jahre 2000 diese Technologie unter dem Namen „AREZZO“ kommerzialisiert.

Idee/Ziel

Idee sowie Ziel von Proforma ist die Bereitstellung eines einfach zu verstehenden Modells zur Entwicklung von Leitlinien mit nur 4 Sprachelementen, die mit zusätzlichen Attributen versehen werden können um eine Entscheidungsfindung zur erleichtern [20].

Sprachelemente

PROforma modelliert Leitlinien als Netzwerk von Tasks, welche an Flussdiagrammen orientiert sind, bestehend aus gekapselten Prozeduren, die bestimmte Ziele verfolgen [24].

Es existieren 4 verschiedene Arten von Tasks:

- Aktionen (Action)
- Pläne (Plan)
- Entscheidungen (Decision)
- Prüfungen (Enquiry)

Aufbau des PROforma Komponentenmodells

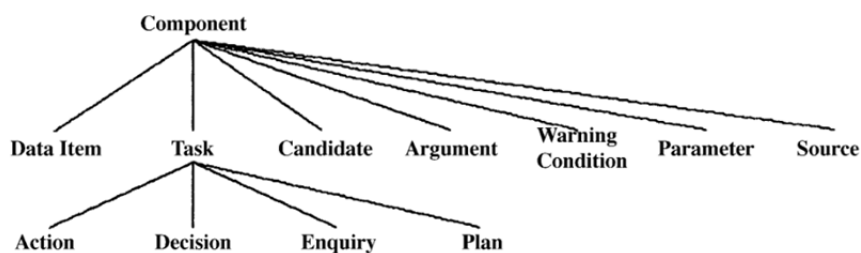


Figure 6. The PROforma component set.

Abbildung 34 : PROforma Komponentenarchitektur

PROforma „components“ bestehen aus einer Anzahl verschiedener Klassen, die in einer Hierarchie angeordnet sind(siehe Bild oben).

- Parameter: Hier erhalten die „Parameter“ einen Datensatz oder Wert zu einem spezifischen Data Item.

<pre>data :: Pregnant ; type :: boolean ; true_value :: Yes ; false_value :: No ; end data .</pre>	<p>Dieses Data Item beschreibt „Schwangerschaft“ mit einem booleschen Datentyp und dessen möglichen Parametern</p>
--	--

- Sources : sind Informationen die von Enquirys angefordert werden
- Warning Conditions : sind Bedingungen die geprüft werden sollten bevor die Daten einem Data Item bereitgestellt werden
- Argument: Enthalten Informationen/Werte zu einem Candidate (z.B. Körpertemp. >35)
- Candidate : Entscheidungsoptionen für eine Decision

Beispiel für Decision, Argument und Candidate

<pre>decision :: Cause ; choice_mode :: multiple ; support_mode :: symbolic ; candidate :: Asthma ; argument :: for, (CXR_result = Normal) ; recommendation :: Netsupport(Cause, Asthma) >= 1 ; candidate :: PNDs ; argument :: for, (CXR_result = Normal) ; recommendation :: Netsupport(Cause, PNDs) >= 1 ; candidate :: GERD ; argument :: for, (CXR_result = Normal) ; recommendation :: Netsupport(Cause, GERD) >= 1 ; candidate :: other ; argument :: for, (CXR_result = other) ; recommendation :: Netsupport(Cause, other) >= 1 ; end decision .</pre>	<p>Dieser Codeauschnitt zeigt eine Decision mit deren möglichen Auswahlmöglichkeiten und den dazugehörigen Argumenten. Zu der jeweiligen Auswahl wird eine bestimmte Empfehlung gegeben.</p>
---	--

- Data Item : Besteht aus Parametern zu einem spezifischen Task (siehe Parameter)
- Task :
 - o Decision: Besteht aus Candidates, die verschiedene Pfade darstellen
 - o Enquiry : Wie bei GLARE eine Anforderung von Daten der Außenwelt.
 - o Plan : Spezifischer Pfad, bestehend aus verschiedenen Tasks
 - o Action : Beschreiben Abläufe die in einem externen Umfeld erledigt werden müssen (z.B. Aktualisieren einer papierbasierten Krankenakte)

Für jeden Task können Attribute definiert werden, wie z.B. Vor und Nachbedingungen. PROforma selbst kombiniert logische Programmierung mit Objektorientierten Design. PROforma wird ebenfalls in vielen kommerziellen Produkten eingesetzt, deshalb stehen keine detaillierten Informationen zu den Ausführungseinheiten zur Verfügung. Produkte basierend auf PROforma sind z.B. : ORAMA,Retrogram,MACRO,ARNO,RAGs,ERA,CAPSULE,e-ASMA,CREDO,HOMEY,LISA

Auf ein Beispiel wurde hier aus Platzgründen verzichtet, kann aber in der Anlage A4 gefunden werden.

Werkzeugunterstützung

Für die Entwicklung von PROforma Leitlinien stehen keine frei zugänglichen Werkzeuge zur Verfügung.

Aktuelle Entwicklungen

PROforma wird als adäquate Sprache zur Codierung von Leitlinien beschrieben. Jedoch ergab sich durch nationale Konventionen zu späteren Zeitpunkten, dass diese auf UML Aktivitätsdiagramme umgestellt werden mussten um sie in weiterführenden Projekten verwenden zu können [99]. Ebenso wurden nur kommerzielle Systeme wie Arezzo weiterentwickelt. Von diesem Produkt sind jedoch keine genaueren Informationen zum genutzten Modell verfügbar.

Arezzo

Formelle Grundlage von Arezzo ist die Modellierungssprache PROforma [46].

„*Arezzo*™ is a commercial product to create, visualize and enact *PROforma* guidelines developed at Cancer Research, UK (2002)“ [23] .

Inzwischen ist Arezzo in Neuseeland eine sehr häufig genutzte Softwarekomplettlösung für Allgemeinmediziner [59; 60]. In England wird ebenso von NHS Direct eine Internetseite auf Basis von Arezzo betrieben. Mithilfe von sogenannten Symptomcheckern kann eine Person dort Selbstdiagnosen durchführen.

3.5 Vorstellung der Domänenneutralen Sprachen

3.5.1 BPMN

Historie

BPMN wurde 2001 von dem IBM Mitarbeiter Stephen A. entwickelt und wurde 2004 von der Business Process Management Initiative (BPMI) veröffentlicht. Die BPMI ist eine Organisation die Standards im Bereich der Geschäftsprozessmodellierung definiert. Mit BPMN 2.0 wurde 2011 der nächste Standard von der Object Management Group (OMG) verabschiedet. Sie erweiterte BPMN um weitere Aspekte der Geschäftsprozessmodellierung.







Idee/Ziel

Idee von BPMN ist die Schaffung einer Geschäftsprozessmodellierungssprache mit einer anschaulich und einfach zu verstehenden Notation.

Ziel von BPMN ist die Geschäftsprozessmodellierung mit der Ausrichtung der Abbildung auf ausführbare Geschäftsprozesse.

Sprachelemente

Nachfolgende Tabelle zeigt eine Übersicht der Sprachelemente der BPMN

Element	Notation
Pool : Ein Pool repräsentiert einen Teilnehmer in einem Prozess und dient der Partinionierung von einer Anzahl von Activities und ebenso als Lauflinie eines Teilnehmers in einem graphischen Container	
Lane : Ist ein Unterelement innerhalb eines Pools und wird der Kategorisierung sowie Organisierung von Activities genutzt	
Data Objects : Diese Elemente haben keinen direkten Einfluss auf den Kontrollfluss eines Prozesses, enthalten aber Informationen darüber was von Activities benötigt, ausgeführt oder erstellt wird	
Group : Dient der Gruppierung von Activities und hat keinen Einfluss auf den Kontrollfluss des Prozesses. Das Group element kann zur Dokumentation oder Analysezwecken eingesetzt werden. Ebenso besteht die Möglichkeit der Gruppierung mehrerer Aktionen eines Prozesses über Pools hinweg	
Text Annotation : Ist ein Mechanismus für den Modellierer zusätzliche Informationen für den Besichtigter eines Prozesses unterzubringen	
Sequence Flow : Wird genutzt um die Reihenfolge bzw. den Ablauf von Activities zu strukturieren	

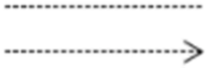




Association : Wird genutzt um Informationen mit Flow Objects zu verbinden	
Message Flow : Wird genutzt um den Informationsfluss zwischen 2 Teilnehmern eines Prozesses zu zeigen, die auf den Empfang sowie das Senden von Informationen eingestellt sind. In BPMN stellen 2 separate Pools 2 Teilnehmer dar.	
Event : Ein Event ist etwas, was während des Prozesses „passiert“ und beeinflussen normalerweise den Ablauf eines Prozesses und bestehen normalerweise aus einem Grund(trigger) oder einem Resultat (result). Der innere Teil der Kreise dient dabei der Differenzierung verschiedener trigger oder results.	
Activity : Ist ein generischer Ausdruck für Aktivitäten die ausgeführt werden müssen. Eine Activity kann ein einzelner (atomarer) Schritt sein oder aus mehreren Activities (compound) bestehen	
Gateway : Gateways werden genutzt um den Kontrollfluss des Prozesses zu beeinflussen. Dies beinhaltet das Aufspalten oder zusammenführen von Prozessen sowie die Wahl von einer Anzahl von Möglichkeiten. Die Art des Gateways wird dabei durch die im inneren des Gateways gegebene Symbolik bestimmt	

Abbildung 35 : BPMN Sprachelemente [110]

BPMN in der Medizin

Wie D Andrzejewsk, P. Ledwon und E Beck 2012 in ihrem Werk [6] zeigten, wurden in Zusammenarbeit mit dem Brandenburgischen Brustzentrum einige derer klinischen Prozesse in BPMN modelliert. Ergebnis war, dass BPMN ein hohes Potential zur Modellierung medizinischer Prozesse aufweist. Ebenso konnten sie zeigen , dass BPMN als Modellierungssprache zur Erstellung von Leitlinien und späteren teilweisen Umwandlung in die Arden Syntax genutzt werden kann [6].

Ein weiterer Ansatz wird von Kristian Duske und Richard Müller beschrieben [55] die am Charite Berlin den Prozess der Schlaganfallbehandlung in BPMN modelliert haben. Diese dienen nach ihrer Aussage als Grundlage für Erweiterungen und künftige Nachforschungen in diesem Gebiet. Beispielhaft wird das Schlaganfallszenario des Charité Berlin nachfolgend dargestellt :

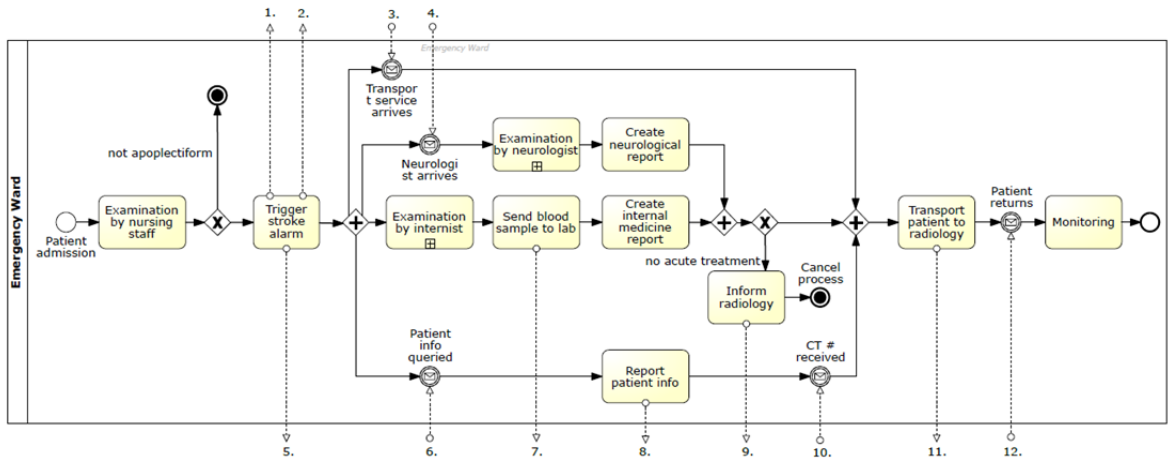


Abbildung 36 : BPMN Ausschnitt für Schlaganfälle in der Notaufnahme im Charité Berlin von [55]

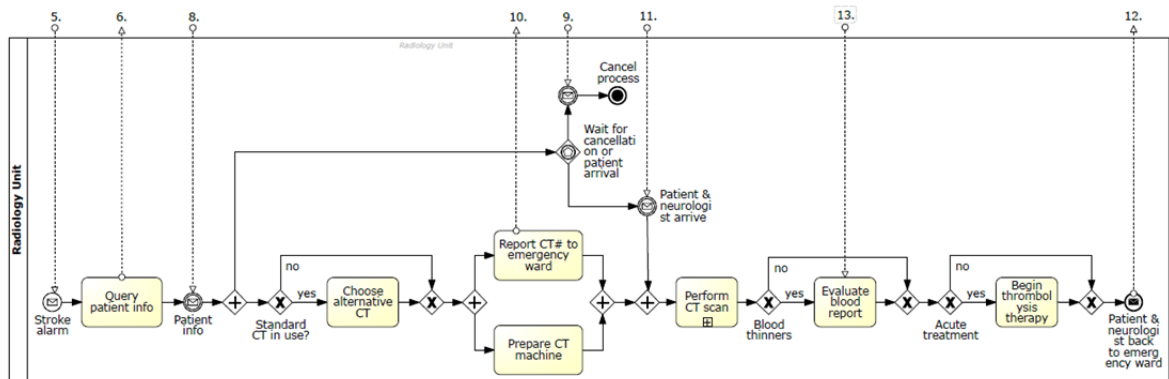


Abbildung 37 : BPMN der Abteilung für Radiologie für Schlaganfälle im Charite Berlin [55]

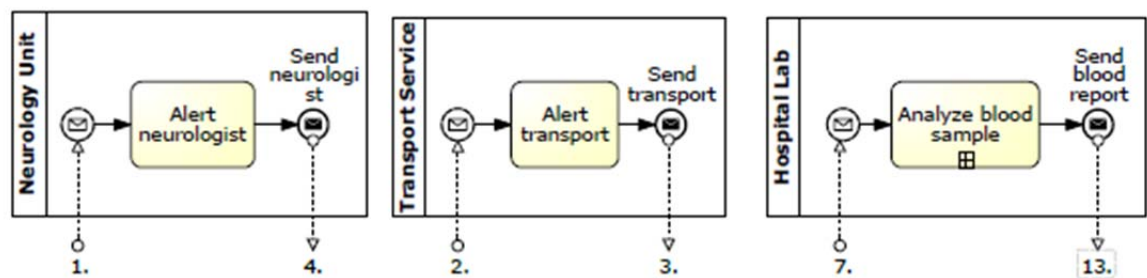


Abbildung 38 : BPMN Abteilung Neurologie, Transport, Labor für das Schlaganfallsszenario [55]

Das obige Beispiel zeigt die Behandlung von Schlaganfällen im Berliner Charité Krankenhaus. Beispielhaft werden die Message Flows 1,3 und 5 ausgelöst, wenn das zuständige Personal bei der Untersuchung des Patienten einen Verdacht auf einen Schlaganfall hat. Durch Telefonate des Pflegepersonals werden 3 andere Ereignisse in den Abteilungen Radiologie, Neurologie und Krankentransport ausgelöst, die dann wieder auf den Hauptprozess zurücklaufen.

Patrik Erklund und Mats Johansson verfassten zu BPMN auch ein erstes theoretisches Werk [31] die sich ausgehend von schlimmen Unfällen im Straßenverkehr mit dem Informationsmanagement in Notfällen über alle beteiligten Organisationen, wie Feuerwehr, Notarzt sowie Polizei mittels BPMN befassen.

Werkzeugunterstützung

Für BPMN existiert eine Vielzahl von freien sowie kommerziellen Werkzeugen. Beispielhaft seien hier ARIS als kommerzielles sowie Yaoqiang BPMN Editor als freies Werkzeug zur Erstellung von generischen BPMN Prozessen genannt.

Aktuelle Entwicklungen

Die OMG veröffentlichte im Januar 2011 die Version 2.0 der BPMN. Im Juli 2013 wurde BPMN in der ISO/IEC 19510:2013 zum internationalen Standard erhoben.

3.5.2 EEPK –erweiterte ereignisgesteuerte Prozesskette

Historie

Die EEPK wurde 1992 in der Universität des Saarlandes in Saarbrücken im Rahmen eines Forschungsprojektes mit der SAP AG entwickelt.

Idee/Ziel

Die erweiterte ereignisgesteuerte Prozesskette (EEPK) ist eine semiformelle Modellierungssprache die nach der ARIS Methode auf der Steuerungsschicht aufsetzt. Die erweiterte ereignisgesteuerte Prozesskette wird dabei um die Daten-, Organisations- und Leistungsmodellierung erweitert.

Nach Staud [17] wird bei der Entwicklung von Modellen die erweiterte ereignisgesteuerte Prozesskette für die Modellierung von Pfaden verwendet und diese, falls Notwendig um die Symboliken der DIN66001 [26] erweitert. Als Grundlage für betriebliche Standardsoftware wie z.B. SAP/R3 können diese dann verwendet werden.

Sprachelemente

Sprachkonstrukte der EEPK sind:

Ereignis

Zustand vor oder nach einer Funktion. Ein sechseckiges Symbol stellt dies graphisch dar.

Funktion

Eine Aktion oder Aufgabe die auf ein Ereignis folgt.
Dargestellt durch Rechtecke mit abgerundeten Ecken

Konnektoren

Dienen zum regeln des Kontrollflusses.
Kontrollflusselemente sind UND, ODER, XOR. Dargestellt werden diese durch einen kleinen Kreis mit dem entsprechenden Symbol

Prozesswegweiser

Hinweise auf andere Prozesse, werden durch ein Rechteck symbolisiert hinter dem sich ein Sechseck verbirgt.

Informationsobjekte

Dokumente oder Sonstige Datenspeicher, dargestellt durch ein Rechteck

Organisationseinheiten: Symbolisieren Rollen oder Personen, dargestellt durch eine Ellipse mit einer senkrechten Linie am linken Rand.

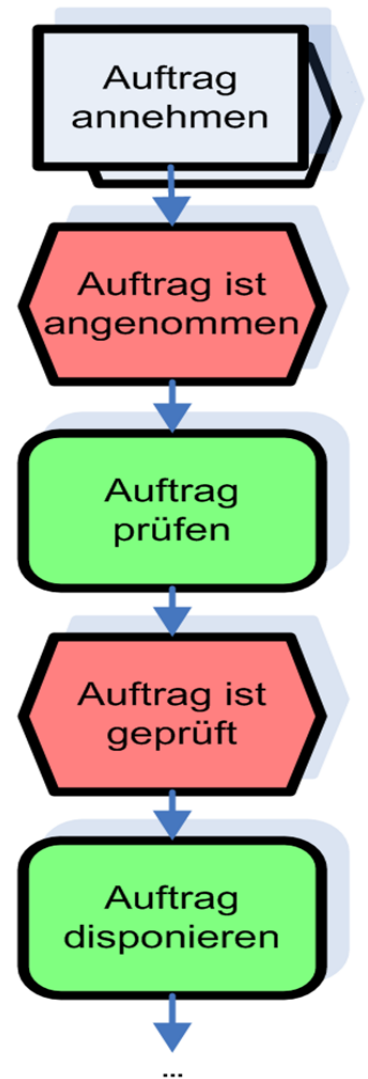


Abbildung 39 : EPK Einfaches Beispiel

EEPK in der Medizin

Eine Verwendung von EEPKs zur Darstellung von Leitlinien oder klinischen Pfaden ist schwer zu finden, da der klinische Algorithmus gleiche Sprachelemente beinhaltet und dieser bevorzugt genutzt wird. Zwar beschreibt z.B. Elliot B. Sloane [32] eine Möglichkeit der Nutzung von EEPKs in Krankenhäusern, diese dienen aber nicht der Leitliniencodierung sondern eher den allgemeinen Arbeitsabläufen in Krankenhäusern.

Werkzeugunterstützung

Im Rahmen der Architektur Integrierter Informationssysteme (ARIS) wird die Methode der sichtenorientierten Modellierung von vielen open-Source sowie kommerziellen Werkzeugen unterstützt. Ebenso ist die Modellierung mittels EEPK in kleinen und mittleren Unternehmen quasi Standard [40]. Zwei bekannte Werkzeuge sind ARIS (Express) [9] oder die auf Eclipse basierende Bflow Toolbox [15]. Ebenso werden für LibreOffice (Extensions [33]) und Microsoft Visio (EPK Shapes [29]) Erweiterungen angeboten.

Aktuelle Entwicklungen

Durch ihre einfache Verständlichkeit erfreut sich die EEPK großer Beliebtheit beim Erlernen von sichtenorientierten Prozessmodellierungssprachen. Als Mittel zur Prozessdokumentation ist sie nach ISO900X (Qualitätsmanagementnorm) [111] ebenso standardisiert und in Verwendung.

3.5.3 UML

UML ist eine auf die Softwareentwicklung ausgerichtete graphische Modellierungssprache die u.a. bei der modellgetriebenen Softwareentwicklung eingesetzt wird. Darüber hinaus kann UML auch zur Prozessmodellierung eingesetzt werden, was es für diese Arbeit interessant macht. In dieser Arbeit werden dafür die Aktivitätsdiagramme aus der UML 2.0 betrachtet, da diese der Leitlinienentwicklung am nächsten kommen.

Historie

Erste Versionen der UML wurden in den 1990er Jahren als Antwort auf zahlreiche Vorschläge für Modellierungssprachen und Methoden der objektorientierten Softwareentwicklung konzipiert. UML wird seit 1997 von der Object Management Group (OMG) als einheitlicher Modellierungsstandard für viele Bereiche der Anwendungsentwicklung vorgeschlagen. UML wurde dabei grundlegend im Jahr 2005 überarbeitet und als UML2 bezeichnet, welche dann in 2008 mit ersten Betaversionen veröffentlicht und 2009 finalisiert wurde.

Idee/Ziel

Idee der Entwicklung von UML war die Vereinheitlichung von Modellierungssprachen und Methoden der objektorientierten Programmierung die in den 1990er Jahren entstanden.

Ziele der UML sind nach [104] :

- Definition einer ausdrucksstarken Modellierungssprache
- Bereitstellung von Erweiterungsmechanismen
- Unabhängigkeit gegenüber Programmiersprachen
- Bereitstellung einer formalen Basis für das Verständnis
- Förderung des Marktwachstums von CASE-Tools
- Unterstützung höherer Entwicklungskonzepte
- Schaffung eines Standards für objektorientierte Modellierungssprachen

UML hatte ursprünglich nicht das Ziel der Entwicklung von Anwendungsprozessen, kann aber durch den Einsatz von Aktivitätsdiagrammen und der Spezifikation des dynamischen Verhaltens des Programmes sinnvoll zur Prozessmodellierung genutzt werden.

Aktivitätsdiagramme können andere Aktivitätsdiagramme oder Teile davon referenzieren um so die Wiederverwendbarkeit zu unterstützen.

Besonderheit dabei ist das Tokenkonzept dieser Diagramme. Diese dienen der verbesserten Darstellung nebenläufiger Abläufe und legt gleichzeitig auch die Abarbeitungsposition des derzeitigen Ablaufs fest. Eine Tokenwanderung durch eine Aktivität bedeutet, dass eine Aktivität bearbeitet wird wobei in einer Aktivität mehrere Token vorkommen können. Diese Token werden im Aktivitätsdiagramm aber nicht graphisch dargestellt.

Sprachelemente

Die UML wird in einer der ersten Ebenen in Spracheinheiten eingeteilt. Diese Spracheinheiten sind eine Menge von zusammengehörigen Modellierungselementen, mit denen ein Modellierer bestimmte Aspekte eines Systems modellieren kann. Nachfolgend werden diese allgemeinen Spracheinheiten für Aktivitätsdiagramme beschrieben, da diese auf die Modellierung von Prozessen ausgerichtet sind.

Name	Beschreibung
Aktion	Ist ein Schritt einer Aktivität in der Benutzer oder Software eine

	Aktion ausführen. Eine Aktion beginnt wenn an allen eingehenden Kontrollflüssen Token angekommen sind. Diese werden nach Bearbeitung der Aktion an alle ausgehenden Flüsse gesendet.
Aktivität	Darunter versteht man eine Kapselung von Aktionen, die mehrere Aktivitäten enthalten können
Kontrollfluss	Eine Verbindung oder Aktivitätskante, die den Kontrollfluss zwischen Aktionen darstellt. Ein Token fließt dabei von einer Aktion zur nächsten.
Objektfluss	Ist eine Verbindung oder Aktivitätskante , über die Objekte von einem Objektknoten zum nächsten fließen können
Objektknoten	Sind Hilfsknoten , die verwendet werden, um den Fluss von Objekten durch das Netzwerk zu spezifizieren
Kontrollknoten	Sind Aktivitätsknoten, die den Kontroll- und Datenfluss einer Aktivität steuern. Diese Umfassen Start-, End-, Parallelisierungs-, Synchronisations-, Verzweigungs- sowie Verbindungsknoten
Partition	Sind mehrere Personen an Aktionen beteiligt geben sogenannte Zuständigkeitsbahnen (Partitions) die Verantwortlichkeitsbereiche der Personen für die Aktionen an
Wächter	Ist eine Bedingung, die angibt , ob ein Token über einen Konnektor übertragen werden darf
Kommentar	Stellt zusätzliche Informationen zu einem Element bereit mit dem es verknüpft ist

UML in der Medizin

Die Prozessmodellierung in der medizinischen Domäne zur Codierung von Leitlinien mittels UML befindet sich derzeit noch in den Kinderschuhen. Wie Publikation von T.Knape [100] oder K. Eisentraut [53] [30] zeigen , wurden prototypisch erste Beispielprozesse in UML Aktivitätsdiagrammen umgesetzt, die aber bisher noch nicht im klinischen Alltag verwendet werden. Zugleich beschreibt T.Knape UML im klinischen Umfeld als enttäuschend [84], da die Integration in bereits vorhandene klinische Systeme schwierig ist. Prinzipiell eignen sich

aber Aktivitätsdiagramme zur Darstellung von klinischen Pfaden. Dazu hat zum Beispiel Hedermann [83] gezeigt, dass in GLIF codierte Leitlinien auch in die UML überführt werden können.

Werkzeugunterstützung

Modelle der UML können in einer Vielzahl (100+) von Werkzeugen entworfen werden, wie [51] zeigt. Beispielhaft sind ARGO UML, Dia, Umbrello, Magic Draw, Umlet und E Violet UML genannt.

Aktuelle Entwicklungen

Die Unified Modeling Language (UML) ist eine Sprache und Notation zur Spezifikation, Konstruktion, Visualisierung und Dokumentation von Modellen für Softwaresystemen [39] und derzeit der de facto Standard in der Modellgetriebenen Softwareentwicklung.

Nach ISO 19505 [50] ist UML auch seit einigen Jahren standardisiert.

Die UML stellt dazu Sprachkonstrukte bereit, die alle konkreten Prozesse darstellen können, unter der Voraussetzung, dass die Notation an die jeweilige Situation/Domäne angepasst wird, z.B. durch Erweiterungen der Notation.

Kapitel 4: Vergleich der Sprachen

4.1 Methodik

Alle vorgestellten Sprachen beruhen auf grundlegenden Formalismen der Modellierung von Prozessen. Die im Grundlagenkapitel gestellten Anforderungen dienen dabei als Grundlage für den nachfolgenden Vergleich.

Nachfolgend werden zunächst allgemeine Sprachkonstrukte gegenübergestellt und deren Sprachelemente für spätere Vergleiche generalisiert auf allgemeine Konstrukte abgebildet, ebenso werden die Möglichkeiten der Ablaufplanung und Einbettung in diesen Sprachen untersucht.

Mithilfe der verallgemeinerten Sprachelemente werden die aus dem Grundlagenkapitel gestellten Anforderungen auf die einzelnen Sprachen angewendet und deren Möglichkeiten gegenübergestellt.

4.2 Allgemeine Sprachelemente

Zuerst soll ein Überblick über die Repräsentationsformalismen der Sprachen gegeben werden.

Die verschiedenen Sprachen verwenden bei ihrer Spezifikation verschiedene Ausdrücke für gleiche logische Sprachkonstrukte. Für die Vergleiche der Sprachen werden hier zunächst die spracheigenen Konstrukte auf die zentralen Konzepte der Prozessmodellierung abgebildet, wie sie in Anforderung 1 des Grundlagenkapitels beschrieben werden. In den nachfolgenden Vergleichen werden dann diese Sprachkonstrukte gegenübergestellt. Folgende Tabelle gibt eine Aufschlüsselung über die verschiedenen Benennungen der Konstrukte der einzelnen Sprachen:

Leitlinienmodell	Aktionen	Entscheidungen	Patientenzustand	Ausführungsstatus
Arden	Action slot	Logic slot	-	-
ASBRU	Plan	Precondition Preference	Temporal patterns	Plan state
EEPK	Funktion	Konnektor	-	-
EON	Action, Consultationaction	Decision	Scenario, activity state	_*
GEM	Action Step	Conditional Step	-	-
GLARE	Action	Decision	-	-
GLIF	Action	Decision	Patient state	_*
GUIDE	Task,Wait, Monitor	Deterministic decision, Nondeterministic decision	Implizit im Petrietz	NA
Klinischer Algorithmus	Tätigkeit	Entscheidung	Klinischer Zustand (nur zu Beginn)	-
Prestige	Protocol	State transition	-	Procedure state
Prodigy	Action, Consultationaction	Decision	Patient Scenario	NA

PROforma	Action, Enquiry	Decision	-	Task state
BPMN	Activity	Gateway	-	-
UML	Action	Decision	-	-

Tabelle 5 : Grundelemente der Sprachen (NA : Keine Informationen verfügbar | * EON und GLIF haben einen Ausführungsstatus, sind aber nicht teil des Repräsentationsmodells, sondern Teil der Ausführungseinheiten) | Teilweise adaptiert nach [93]

In der obigen Tabelle ist zu erkennen, dass alle Sprachen die Möglichkeit besitzen grundlegende Sprachelemente wie Aktionen sowie Entscheidungen zu modellieren. Die Modellierung des Patientenzustandes sowie des Ausführungsstatus ist nur in Asbru möglich.

Prodigy, GLIF sowie EON besitzen nur die Möglichkeit den Patientenzustand zu modellieren. GUIDE und Klinischer Algorithmus bilden dabei noch eine Ausnahme, da die Patientenzustände nur in besonderen Fällen verfügbar sind und nicht jederzeit innerhalb einer Leitlinie modelliert werden können.

PROforma und Prestige bilden die Gruppe von Sprachen, die keine Möglichkeit der Modellierung des Patientenzustandes besitzen, jedoch ist der Ausführungsstatus modellierbar.

Die Sprachen PROforma, Prodigy, Prestige, Klinischer Algorithmus, GUIDE, GLIF und EON besitzen so nur die Möglichkeit der Modellierung eines der beiden Konstrukte.

Arden, GEM und GLARE sowie die domänenneutralen Sprachen BPMN, UML und EPK bieten mit ihrem Standardmodell keine Möglichkeit der Modellierung von Patienten- sowie Ausführungsstatus.

4.3 Ablaufplanung und Einbettungsmöglichkeiten der Prozesse

Da verschiedene Ansätze der Repräsentation der Prozesse existieren, zeigt die folgende Tabelle einen Überblick über die Aspekte der Leitliniendarstellung. Betrachtet werden die Art der Ablaufplanung bzw. die Darstellungsform, die die möglichen Reihenfolgsbeziehungen der Prozessschritte beschreibt. Möglichkeiten der Einbettung von Subplänen und der Möglichkeit Patientendaten zu modellieren wird hier ebenso untersucht.

Leitlinienmodell	Ablaufplanung	Einbettung von Leitlinien	Patientendaten
Arden	Modul Invocation	n.m.	n.m.
ASBRU	Plan-body	Plan	n.m.
EEPK	Flussdiagramm	n.m.	n.m.
EON	Flussdiagramm	Subguideline	EMR Ontologie
GASTON	Flussdiagramm	Subguideline	Domain Ontologie
GLARE	Flussdiagramm	Composite Action	electronic patient record
GLIF	Flussdiagramm	Subguideline	Domain ontology
GUIDE	Flussdiagramm	Task	Relation
Klinischer Algorithmus	Flussdiagramm	n.m.	n.m.
Prestige	Zustandsübergangsdiagramm	Protocol	Patient record model
Prodigy	Zustandsübergangsdiagramm	Subguideline	EMR Ontologie
PROforma	Constraint satisfaction graph	Plan task	Patent data definition
BPMN	BPMN Prozessdiagramm	Composite	-
UML	Aktivitätsdiagramm	Nested classifier	-

Abbildung 40 : Ablaufplanung, Einbettung und Patientendaten | n.m. = nicht möglich

Durch den speziellen Aufbau der Arden MLMs gibt es keine Möglichkeit der Einbettung von Teilen einer Leitlinie in ein Arden MLM. Arden verfolgt dabei ein anderes Prinzip und fasst viele einzelne MLMs zu Paketen (MKP – Medical Knowledge Packages) eines zugehörigen Fachbereichs zusammen. Zugleich sind noch EEPK und Klinischer Algorithmus nicht in der Lage eine Einbettung weiterer zusammengehöriger Schritte zu modellieren.

Alle anderen Sprachen besitzen unterschiedliche Möglichkeiten der Einbettung von zusammengehörigen Behandlungsschritten.

Die Modellierung von Patientendaten erfolgt von allen Sprachen, die nicht mit „n. m.“ markiert sind, Abstrakt in Ontologien die dann auf die Konstrukte der darunterliegenden Informationssysteme abgebildet werden können. Die Sprachen mit „nicht möglich“ rufen zur

Laufzeit einzelne Datenelemente zu einem Patienten ab und verarbeiten diese sofort, ohne sie für weitere Schritte zu verwenden.

Bei den Domänenneutralen sprachen UML und BPMN existieren keine Sprachkonstrukte für die Modellierung der Patientendaten, jedoch stehen standardisierte Sprachkonstrukte zur Spezifikation der Patientendaten zur Verfügung.

4.4 Anwendung der Anforderungen aus dem Grundlagenkapitel

Alle Anforderungen an die Sprachen werden hier noch einmal genauer untersucht, da vorhandene Quellen unterschiedliche oder nicht verständliche Angaben zu bestimmten Sprachelementen machen.

4.4.1 Anforderung 1

Alle Sprachmittel, die in Anforderung 1 genannt werden, lassen sich in allgemein vergleichbare Konstrukte der Prozessmodellierung, wie in Tabelle 9 dargestellt, überführen. Dabei wird erkenntlich, dass alle Sprachen die Möglichkeit besitzen Behandlungsschritte, Entscheidungen und Prozessabläufe zu beschreiben.

Eine Modellierung des Patientenzustandes ist hingegen nur in EON, GLIF, GUIDE, Proforma und Prodigy möglich. Der klinische Algorithmus bildet dabei noch eine besondere Ausnahme, da dort eine Modellierung des Patientenzustandes nur am Anfang aber nicht innerhalb des Prozesses möglich ist.

Entscheidungen sind zusätzlich bei GLIF, GLARE und GUIDE in 2 Kategorien unterteilt, in denen eine Kategorie evidenzbasiert und die andere subjektiv vom Arzt entschieden wird.

4.4.2 Anforderung 2

Die in Anforderung 2 beschriebenen Prozesskontrollstrukturen für variable Abläufe sind in keinem der Ansätze bis auf ASBRU eindeutig vorgegeben. Durch seine Zeitnotation ist es in ASBRU möglich einen variablen sowie parallelen Ablauf von Behandlungsschritten zu modellieren, die ausgehend von einem Referenzzeitpunkt und Intervallen für eine minimale

und maximale Behandlungsdauer dargestellt werden können. Die EEPK, BPMN, UML, GLARE und GUIDE nutzen logische Konnektoren um parallele Abläufe zu Modellieren. Arden kann mithilfe der Definition mehrerer auszuführender Pläne im Action Slot ebenso eine Parallelisierung realisieren. GLIF, GEM sowie EON besitzen die Möglichkeit mittels des Branch Steps eine Parallelisierung zu realisieren.

4.4.3 Anforderung 3

Konzepte zur Modellierung von Ressourcen und Organisationseinheiten aus Anforderung 3 werden von GLIF, GUIDE und PROforma nur teilweise unterstützt die durch Festlegung von zusätzlichen Attributen modelliert werden können. Die EEPK ermöglicht die beste Integration von Verantwortlichkeiten und beteiligten Organisationseinheiten mittels der Sprachelemente Informationsobjekt und Organisationseinheit. BPMN ermöglicht die Modellierung von Ressourcen mittels Resources sowie die Modellierung von Organisationseinheiten mittels Pools und Lanes. UML bietet ähnlich zu BPMN die Einteilung in Organisationseinheiten mittels der Partitions, jedoch ist eine Modellierung von geschäftsprozessrelevanten Ressourcen in UML nicht vorgesehen.

4.4.4 Anforderung 4

Die in Anforderung 4 geforderten Konzepte für evidenzbasierte Entscheidungen werden teilweise von Arden, ASBRU, GLARE, GLIF, GUIDE und PROforma unterstützt. Diese Sprachen können bei der Ausführung notwendige Informationen von Umgebenden Systemen oder mittels Nutzereingaben zur Entscheidungsfindung nutzen.

Arden ermöglicht eine Entscheidung mit Hilfe der im Knowledge Slot definierten Informationen, wie z.B. einem zuletzt gemessenen Blutwert. GUIDE, GLARE, PROforma und GLIF unterscheiden regelbasierte und subjektive Entscheidungen, bieten aber keine umfassende Möglichkeit der evidenzbasierten Entscheidung. Subjektive Entscheidungen werden dabei nicht vom Softwaresystem entschieden, sondern vom behandelnden Arzt, der je nach Sprache zusätzliche Informationen zu einer subjektiven Entscheidung in Form von weiterführenden Informationen und Links erhält.

In ASBRU können mithilfe von Absichten Entscheidungshilfen gegeben werden. Diese Absichten beschreiben mit der Angabe einer Wahrscheinlichkeit ein Behandlungsziel, das

aussagt welchen Effekt ein bestimmter Behandlungsschritt oder die Verabreichung eines Medikament auf den Patient haben müsste, bieten aber keine eigentliche Unterstützung für evidenzbasierte Entscheidungen. Das in ASBRU verwendete Konzept der Absichten kommt den evidenzbasierten Entscheidungen dabei am nächsten.

4.4.5 Anforderung 5

Die Integration von den in Anforderung 5 geforderten Kenngrößen für ökonomische und zeitliche Kenngrößen wird von keiner der Sprachen bis auf teilweise Asbru ermöglicht. Asbru bietet dabei nur die Möglichkeit der Modellierung von Kosten für Behandlungsschritte, jedoch können keine Erlöse für die Behandlungsschritte modelliert werden. Die zeitliche Terminierung von Asbru Plänen wird dabei über die maximale Behandlungsdauer mithilfe des Referenzzeitpunktes modelliert. GLARE bietet bei subjektiven Entscheidungen die Möglichkeit der Einbeziehung von Kosten sowie Effektivitätskriterien und Nebenwirkungen, diese können aber nur bei der jeweiligen Entscheidung modelliert werden und bietet keine allgemeine Lösung der Modellierung der in Anforderung 5 genannten Aspekte über eine gesamte Leitlinie.

Anforderung	A1				A2		A3	A4	A5
Sprache	Patientenzustand	Behandlungsschritt	Entscheidung	Prozessfluss	Variabler Ablauf	Parallelisierung	Ressourcen, Organisationseinheiten	evidenzbasierte Entscheidung	Ökonomische Kenngrößen
Arden	-	X	X	X	-	O	-	O	-
ASBRU	-	X	X	X	X	X	-	O	O
EEPK	-	X	X	X	-	X	X	-	-
EON	X	X	X	X	-	X	-	-	-
GEM	-	X	X	X	-	X	-	O	-
GLARE	-	X	X	X	-	X	-	O	O

GLIF	X	X	X	X	-	X	O	O	-
GUIDE	X	X	X	X	O	O	O	O	-
Klinischer Algorithmus	O	X	X	X	-	-	-	-	-
Prestige									
Prodigy	X	X	X	X	-	-	-	-	-
PROforma	-	X	X	X	-	-	-	O	-
BPMN	-	X	X	X	-	X	X	-	-
UML	-	X	X	X	-	X	O	-	-

Abbildung 41 : Anforderungserfüllung der Modellierungsansätze adaptiert nach [63] | X Anforderung erfüllt | O Anforderung Teilweise erfüllt | - Anforderung nicht erfüllt | qu = Quellenlage unzureichend.

Die obige Tabelle zeigt eine Zusammenfassung der Anforderungen 1-5 an klinische Modellierungssprachen. Es ist zu sehen, dass keine der Modellierungssprachen alle Anforderungen erfüllt.

Kapitel 5: Ergebnisse

Zu Beginn sei darauf hingewiesen, dass keine der Sprachen alle Anforderungen der von Martin Burwitz, Hannes Schlieter und Werner Esswein [63] sowie Shiffman [87] genannten Anforderungen erfüllt. Auf diese Anforderungen wird später noch einmal genauer eingegangen. Zuerst werden aber die von Shiffman beschriebenen Anforderungen an Leitlinienmodellierungssprachen genauer für die domänenneutralen Sprachen erläutert.

5.1 Domänenneutrale Sprachen

Zur Bewertung der Domänenneutralen Sprachen werden die von Shiffman et al. [88] genannten Anforderungen aus dem Grundlagenkapitel auf diese Angewendet. Dabei wurde der klinische Algorithmus mit zu diesen Sprachen gezählt, da bis auf die Modellierung eines Anfangszustandes des Patienten keine Domänenspezifischen Sprachkonstrukte existieren.

Anforderung/Sprache	EEPK	BPMN	UML	Klinischer Algorithmus
Umfassend	-	-	-	-
Angemessene Ausdrücke	X	X	X	X
Flexibel	-	X	X	-
Verständlich	X	X	X	X
Austauschbar	X	X	X	X

Tabelle 6 : Anforderungen von Shiffman an Sprache zur Modellierung von Leitlinien | X = Anforderung erfüllt | - = Anforderung nicht erfüllt

Diese Tabelle zeigt eine kurze Übersicht der von Shiffman gestellten Anforderungen an Sprachen zur Modellierung von Leitlinien, wobei eine scharfe Abgrenzung der Anforderungen nicht möglich ist. Eine genaue Beschreibung erfolgt deshalb im folgenden Kapitel.

Domänenneutrale Modellierungssprachen sind für den Einsatz in der medizinischen Problem-domäne unterschiedlich gut geeignet. EEPK und klinischer Algorithmus werden dabei zusammen betrachtet, da diese sich nur gering voneinander unterscheiden.

EEPK und der klinische Algorithmus sind dabei am problematischsten einzustufen, da sie nur aus einer geringen Zahl von Sprachelementen bestehen und nicht anpassbar sind, welches einen praktischen Einsatz erschwert. Genau diese Einschränkung stellt aber auch einen Vorteil sowie Nachteil dieser Sprachen dar, da sie dadurch bei weniger komplexen Prozessen einfach zu verstehen sind, aber durch die wenigen generischen Sprachkonstrukte müssen domänenspezifische Sachverhalte aufwendig nachgebildet werden, welches die modellierten Sachverhalte künstlich aufbläht und unübersichtlich macht. Dies stellt somit auch einen erheblichen Nachteil gegenüber den Domänenspezifischen Sprachen dar. Eine Operationalisierbarkeit von EEPK und klinischen Algorithmus ist dabei auch ungenügend. Diese Modelle sind somit nur mit einer nachträglichen Umwandlung in andere Modellbasierte Ansätze verwendbar. Eine Modellierung des Patientenzustandes ist nur am Anfang eines klinischen Algorithmus möglich, was einen Bezug auf den Patientenzustand während der Behandlung erschwert. Der Einbezug von Ressourcen ist in der EEPK nicht

eindeutig und im klinischen Algorithmus nicht möglich. Ressourcen wie Diagnosegeräte oder Operationssäle sind in einem Krankenhaus nur in begrenzter Anzahl verfügbar, worauf weder die EEPK noch der klinische Algorithmus Rücksicht nimmt. Bei der EEPK dient die Modellierung von Organisationseinheiten und Ressourcen auch eher der Strukturierung und Zuordnung von Verantwortlichkeiten. Eine Modellierung von konkurrierenden Ressourcen ist somit nicht möglich. Die Ausdrucksmächtigkeit ist für medizinische Sachverhalte nur bedingt geeignet, wichtige Sachverhalte können aber abgebildet werden. Insgesamt erscheint die Modellierung von Leitlinien oder klinischen Pfaden mit EEPK und klinischem Algorithmus nur bedingt geeignet, da wichtige Anforderungen nicht erfüllt werden. Jedoch hat sich durch die einfache Notation die Darstellung von wichtigen Teilen einer Leitlinie als klinischen Algorithmus durchgesetzt, da diese wie erwähnt einfach zu verstehen sind. Erstellte Modelle sind dabei über Institutionen hinweg austauschbar, da keine institutionsspezifischen Informationen einfließen.

BPMN ist mit der Ausrichtung auf Geschäftsprozesse ebenso bedingt geeignet für eine standardisierte Darstellung von Leitlinien oder klinischen Pfaden, wobei eine Ausrichtung auf eine spezielle Zieldomäne, z.B. die der Medizin, realisierbar ist. Dies wird durch das vorhandene Metamodell ermöglicht, welches durch intelligente Anpassung des Modells an die medizinische Domäne wie Andrzejewski [6] zeigt erfolgt. Dies erfordert aber mitunter auch eine Erweiterung der Sprache um zusätzliche Elemente. Somit könnten alle Wissens Elemente einer Leitlinie dargestellt werden. Die Ausdrucksmächtigkeit ist somit ähnlich zur EEPK als gut einzuordnen, da alle wichtigen Elemente abgebildet werden können. Die Notation von BPMN schneidet in Anwenderbezogenen Kriterien gut ab und ist einfach zu erlernen und anschaulich. Durch Kapselung von Aktionen in Activities ist es möglich verschiedene Granularitätsstufen darzustellen, welche Verständlichkeit und Übersichtlichkeit der Sprache stark erhöhen.

Da bisher wenig dokumentierte Erfahrungen zu BPMN im medizinischen Sektor vorliegen ist eine Bewertung aber dennoch nicht einfach, jedoch konnte gezeigt werden, dass BPMN als Modellierungssprache zur Erstellung von klinischen Pfaden oder Leitlinien und späteren Umwandlung in den Arden Syntax genutzt werden kann [6]. Zugleich können die BPMN Prozesse wie von Kristian Duske [55] am Charité Berlin gezeigt aber auch direkt genutzt werden. Durch die standardisierte Ausführereinheit von BPMN (BPEL) wird ein

Institutsübergreifender Austausch stark vereinfacht, wobei Institutionsspezifische Besonderheiten der Leitlinie aber gleich bleiben müssen, wie z.B. das Vorhandensein bestimmter Fachabteilungen oder Diagnosegeräte.

UML hat ebenso wie BPMN und die EEPK vor und Nachteile was deren Verwendbarkeit in der medizinischen Problemdomäne angeht. Wie im Hauptteil erwähnt besteht über das gegebene Metamodell die Möglichkeit der Erweiterung der UML um domänenspezifische Konstrukte mit dem Nachteil, das die Verständlichkeit sowie Lesbarkeit des Aktivitätsdiagrammes leidet. Zugleich ist die Anpassung der UML mit nicht unerheblichen Aufwänden verbunden, die dann jedoch wiederverwendbar wären. Eine Abbildung von Ressourcen und organisatorischen Aspekten ist nur schwer möglich und standardmäßig von der UML nicht vorgesehen. Dies ist einen Nachteil in Bezug auf eine optimale Ressourcennutzung in der medizinischen Domäne. Einer der größten Vorteile ist die gute Operationalisierbarkeit. Durch die Möglichkeit der Referenzierung von Teilen oder ganzen Aktivitätsdiagrammen besteht die Möglichkeit der Darstellung von unterschiedlichen Granularitätsstufen eines Planes, was die Übersichtlichkeit und Wiederverwendbarkeit erhöht. Die Anwendung von Aktivitätsdiagrammen ist leicht zu erlernen, jedoch existieren, durch den generischen Ansatz, keine domänenspezifischen Modellierungskonstrukte. Die Verständlichkeit lässt daher aus Anwendersicht zu wünschen übrig, da spezielle Konstrukte fehlen.

UML kann durch seine Erweiterbarkeit jedoch mithilfe von Profilen und Stereotypen um diese Konstrukte erweitert werden um die Verständlichkeit und Lesbarkeit deutlich zu verbessern. Diese Erweiterbarkeit ist auch ein Vorteil in Bezug auf die medizinische Problemdomäne. Generell können die Schwächen durch die Erweiterungsmöglichkeiten der UML durch entsprechende Konstrukte weitgehend überwunden werden, jedoch stellt sich dann die Frage nach der Verhältnismäßigkeit.

Damit ist festzustellen, dass die UML bei der Modellierung umfangreicher Prozesse mit Aktivitätsdiagrammen durch die umfangreichen Anpassungen, die erforderlich sind, nur bedingt geeignet ist, obwohl eine sehr gute Formalisierung und Operationalisierbarkeit von Vorteil sind.

5.2 Domänenspezifische Sprachen

Bei den domänenspezifischen Sprachen kann zuerst eine Unterteilung in objektorientierte Sprachen und nicht objektorientierten Sprachen gemacht werden wie die nachfolgende Übersicht zeigt.

Typ	Objektorientiert	Nicht objektorientiert
Sprachen	EON GEM GLIF	ASBRU Arden GLARE GUIDE Prodigy PROforma

5.2.1 Objektorientierte domänenspezifische Sprachen

Am schwierigsten gestaltet sich die Einordnung von GEM, da diese Sprache versucht den Spalt zwischen natürlichsprachiger Leitlinie und implementierungsnahen Sprachen zu schließen. GEM ist dabei auf die englische Sprache ausgelegt und kann sich schlecht auf die Komplexität deutscher Sprachkonstrukte einstellen, welche weitaus mehr Formulierungsvarianten eines bestimmten medizinischen Sachverhaltes unterstützen. Eine umfassende Beschreibung einer Leitlinie in GEM in deutscher Sprache kann somit nicht gewährleistet werden. Die Ausdrucksmächtigkeit von GEM ist als befriedigend einzustufen da alle Nuancen einer Leitlinie abgebildet werden können, obwohl eine Erweiterung der Sprachkonstrukte nicht möglich ist. Diese befinden sich jedoch nicht in einem computerinterpretierbaren Format, da schwierige Leitlinienausschnitte weiterhin als natürlicher Text innerhalb von GEM codiert sind. Die Codierung von natürlichem Text erhöht somit die Verständlichkeit der Leitlinie und ebenso ist die Struktur und der Aufbau von den in GEM codierten Leitlinien immer gleich. Da GEM Leitlinien im Allgemeinen codiert und keine institutionsspezifischen Sachverhalte mit einbezieht, ist ein Institutsübergreifender Austausch dieser Leitlinien möglich und kann als Grundlage für weiter verfeinerte Prozesse genutzt werden. Durch das Alleinstellungsmerkmal der Codierung von natürlichsprachigen Leitlinien in semistrukturierte Dokumente wurde GEM zu einem internationalen ASTM Standard erhoben.

EON sowie dessen Weiterentwicklung GLIF erweitern die Möglichkeiten der objektorientierten Sprache GEM um die Ausführbarkeit, durch Bildung von objektorientierten Ontologien. Beide Sprachen sind umfassend und können alle Sachverhalte einer Leitlinie abbilden. Die Ausdrucksmächtigkeit ist besser als bei EON als gut einzuordnen, da diese Sprachen um weitere Klassen und Attribute erweitert werden können, falls neue Aspekte bei der Leitlinienentwicklung hinzukommen. Jedoch leidet hierunter die Verständlichkeit der darin codierten Modelle, da es sich um eine Vielzahl (100+) von Klassen mit deren Attributen handelt. Eine Austauschbarkeit dieser Leitlinien ist ein vorrangiges Ziel von GLIF, wobei für EON dies keins ist. EON wurde eher als eine Art Middleware entwickelt, dessen Komponenten für die Modellierung von Leitlinien aber nur sehr wenig Bedeutung finden. EON weist daher nur geringe Unterschiede zu GLIF aus der Modellierungssicht auf.

5.2.2 Nicht Objektorientierte domänenspezifische Sprachen

Arden ist eine Besonderheit unter allen bisher genannten Sprachen. In den MLMs, die jeweils einzelne Schritte eines klinischen Pfades umfassen, wird jeweils auch nur eine Aktion beschrieben. Das führt zu einfach zu verstehenden MLMs, da das MLM nur diesen Einzelschritt umfasst. Die Ausdrucksmächtigkeit ist als gut einzuschätzen, da alle wichtigen Aspekte der Leitlinienmodellierung abgebildet werden können. Hierdurch sind diese Einzelschritte leicht wiederbenutzbar und ebenso austauschbar, wenn keine „curly braces“ Ausdrücke innerhalb der MLMs genutzt wurden. Durch speziell in Arden definierte Operatoren ist Arden ebenfalls in der Lage alle wichtigen Teile einer Leitlinie über viele MLMs hinweg zu beschreiben und kann damit auch als umfassend gewertet werden. Durch Medical Knowledge Packages (MKP) können weitere Abstraktionsgrade geschaffen werden, indem viele einzelne MLMs in bestimmte Themen- oder Fachgebiete zusammengefasst werden.

ASBRU bildet ebenso ein Alleinstellungsmerkmal durch das Arbeiten mit Zeitintervallen und Referenzzeitpunkten für einzelne Behandlungsschritte, welche es ihr ermöglicht einen variablen Ablauf von Behandlungsschritten zu modellieren. Die Ausdrucksmächtigkeit ist als gut einzustufen, da alle wichtigen Aspekte einer Leitlinie abgebildet werden können. Im Gegensatz zu den anderen domänenspezifischen Sprachen ist ASBRU als einzigste in der Lage

ökonomische Kenngrößen in den Behandlungsprozess einfließen zu lassen. Arden ist nicht sehr einfach zu verstehen, da die graphischen Darstellungen kein Gesamtbild einer Behandlung geben. Ebenso ist ein Lesen der in XML gespeicherten klinischen Pfade für fachfremde Personen als schwierig einzustufen, da die Pläne mit steigender Komplexität immer mehr suchen innerhalb der XML Datei erfordern, da die Behandlungsschritte nicht nacheinander gespeichert sind. Durch die Speicherung als skeletaler Plan sind diese leicht wiederbenutzbar, zugleich steigt damit auch die Wiederbenutzbarkeit für andere Institutionen.

GLARE, GUIDE, Prodigy und PROforma bilden an Flussdiagrammen orientierte Modelle zur Darstellung ihrer Leitlinien. Durch die geringe Anzahl an Sprachkonstrukten sind all diese Sprachen als gut verständlich einzustufen. Keine dieser Sprachen ist als umfassend einzustufen, da durch die geringe Anzahl von Sprachkonstrukten nicht alle Teile einer Leitlinie abgebildet werden können. Zugleich besitzen alle vier Sprachen die Möglichkeit der Einbettung von verfeinerten Behandlungsabläufen. GUIDE ist dabei als einzigstes System zur Aus- und Weiterbildung entwickelt worden. Prodigy wurde für chronische Krankheiten entwickelt, dadurch ist ein Einsatz über die gesamte medizinische Domäne ausgeschlossen. Die Ausdrucksmächtigkeit der beiden Sprachen ist daher als ausreichend einzuschätzen, da nur wichtigste Grundkomponenten der Leitlinienerstellung existieren. Als Standaloneumgebung wäre Guide aber sehr gut austauschbar und wiederbenutzbar, wobei nie eine Ausführereinheit bereitgestellt wurde.

GLARE und PROforma unterscheiden sich nur gering in der graphischen Darstellung zu GUIDE und Prodigy, wobei diese keinen speziellen Augenmerk auf Aus- und Weiterbildung sowie die Behandlung von chronischen Krankheiten legen.

GLARE nutzt für Entscheidungsmodellierungen ein Punktesystem, welches Punkte bestimmter Parameter aufaddiert umso die Wahl nach der höchsten erreichten Punktzahl zu treffen. Die Ausdrucksmächtigkeit ist aber trotzdem nur als befriedigend einzustufen, da bestimmte Sachverhalte einer Leitlinie nur kompliziert nachgebildet werden können. GLARE nutzt verschiedene Zwischenschichten um auf die Informationen des umgebenden Systems zuzugreifen. Mit einer Anpassung des Mappings dieser Zwischenschicht ist somit eine Austauschbarkeit sowie Wiederbenutzbarkeit mit relativ geringem Aufwand möglich, da keine Leitlinie angepasst werden muss.

Proforma beschreibt nicht explizit wie die Sprache an das umgebende System gebunden wird, somit kann über Austauschbarkeit sowie Wiederbenutzbarkeit keine Angabe gemacht werden.

Kapitel 6: Zusammenfassung und Ausblick

Wie in Kapitel 3 schon fast ersichtlich, werden viele der Sprachen nicht mehr weiterentwickelt oder sie wurden im Laufe der letzten Jahre kommerzialisiert um deren Weiterentwicklung zu gewährleisten. Dadurch ist ein freier Zugriff auf aktuelle Versionen dieser Sprachen meistens nicht mehr möglich. Sprachen die seit mehr als 3 Jahren nicht mehr weiterentwickelt wurden werden als nicht fortgesetzt bewertet. Nachfolgende Tabelle zeigt eine Übersicht:

	Kommerziell	Frei zugänglich	Internes Projekt	Nicht fortgesetzt
Sprache	Arden	Arden	GLARE	ASBRU
	EON	ASBRU	GUIDE	EON
	Prodigy	GEM		GLARE
	PROforma	GLIF		GLIF GUIDE Prodigy Proforma

Tabelle 7 : Aktueller Status der Sprachen

Die obige Tabelle zeigt, dass GEM sowie Arden die einzigsten 2 Sprachen sind, die aktuell gepflegt und weiterentwickelt werden und dabei frei zugänglich bleiben. Alle anderen Sprachen wurden wie z.B. EON, PROforma und Prodigy kommerzialisiert und intern weiterentwickelt oder wurden einfach gar nicht mehr weiterentwickelt. Bei der Wahl nach einer aktuellen Modellierungssprache entfallen eben viele dieser Sprachen durch die genannte Aspekte.

Wie diese Arbeit zeigt, ist keine der vorgestellten domänenspezifischen Modellierungssprachen in der Lage die gestellten Anforderungen wie sie Shiffman oder Schlieter beschreiben zu erfüllen. Bemerkbar ist dabei die Entwicklung der Domänenneutralen Sprachen, deren Erfüllungsgrad der Anforderungen ähnlich zu denen der domänenspezifischen ist. Ein großer Vorteil der domänenneutralen Sprachen ist zusätzlich,

das durch gegebene Metamodelle bei UML und BPMN die Sprachkonstrukte erweiterbar und auf eine spezifische Zieldomäne ausrichtbar sind.

Letztlich ist somit keine der Sprachen auf einem Niveau, die einen einheitlichen Einsatz über die gesamte Medizinische Domäne ermöglicht, sodass noch Forschungsbedarf in diesem Gebiet besteht.

Anlagen

A1

	Ergebnisse	Wichtigste Arbeiten	Zitiert durch	Ergebnisse Zitiert durch	Seit 2010	Seit 2013	Seit 2014
Arden Syntax	5570	Rationale for the Arden syntax	264	252	31	8	3
		The Arden Syntax for Medical Logic Modules. Writing Arden Syntax medical Logic modules	172	161	10	2	2
		Asbru: A task-specific, intention-based, and time-oriented language for representing 906 skeletal plans	109	102	31	8	1
Asbru	906 skeletal plans	Asbru: A task-specific, intention-based, and time-oriented language for representing 906 skeletal plans	151	143	31	6	0
		Formal semantics of Asbru-an overview	37	35	8	2	0
CPG-RA	18	Asbru 7.2 reference manual	38	38	3	0	0
		Practical evaluation of clinical guideline docu Guidelines and workflow models	7	7	4	0	0
EON	18700	Keine Relevanten arbeiten	18	16	7	1	0
GASTON	31600	Design and implementation of a framework tc	85	80	30	9	1
GEM II	17600	Keine Relevanten Arbeiten (55Eiten)					
GEM	17400	GEM: a proposal for a more comprehensive guideline document model using XML	286	258	47	8	2
GLARE	50	Applying artificial intelligence to clinical guide Automatic checking of the correctness of clinical guidelines contextualizator	14	12	8	2	0
		Managing clinical guidelines contextualizator	5	5	1	0	0
		The Guideline Interchange Format A Model fo	3	1	0	0	0
GINSENG	Keine Relevanten , Project begann 2013 . 2016 Geplante Fertigstellung	The Guideline Interchange Format A Model fo	408	376	56	15	4
		GLIF3: the evolution of a guideline representa	290	266	40	9	0
		GLIF3: a representation format for sharable co	188	178	83	19	3
GUIDE	1,7Mio	Keine Relevanten					
HELEN	16400	HELEN, a modular framework for representing 15 Keine Relevant	16	15	7	3	0
		HELEN, a modular framework for representing 15 Keine Relevant	0	0	0	0	0
HGMML	7	Keine Relevanten	0	0	0	0	0
PRODIGY	342						
SAGE	357000	Clinical decision-support systems	144	123	57	17	0
		Keine Relevanten	0	0	0	0	0
SUMMIE	482244		2235	2068	454	109	16

Tabelle 8 : Recherche Sprachen Zusammenfassung

Die Tabelle Zeigt die drei am meisten referenzierten Werke zu den Sprachen. Zitiert durch gibt die Anzahl der Referenzen des Werkes an. Ergebnisse zitiert durch gibt an wie viele

Referenzen tatsächlich zu einem Werk existieren. Diese Ergebnisse wurden dann noch einmal nach Jahr dargestellt um einen kleinen Einblick in die Aktualität der Sprachen zu erhalten. Dies zeigt, wie sich die Sprachen im Laufe der Jahre Entwickelt haben, und ob diese noch praktische Relevanz besitzen könnten.

A2

Die folgende Tabelle zeigt die bei der Literaturrecherche gefundenen Werke zu den jeweiligen Stichwörtern:

Schlagwort	Gefundene Werke
Clinical guideline	<p>Comparing computer-interpretable guideline models: a case-study approach</p> <p>... primitives, process models and patient data in computer-interpretable clinical practice guidelines:: A literature review of guideline representation models</p> <p>Support for interactive verification of asbru in kiv</p> <p>Verification of medical guidelines by model checking—a case study</p> <p>" Kontrollflussttransformation von BPMN zu Asbru"; Betreuer/in (nen): H. Eidenberger, K. Kaiser; Softwaretechnik und Interaktive Systeme, 2013.</p> <p>Asbru: A task-specific, intention-based, and time-oriented language for representing skeletal plans</p> <p>Model checking for clinical guidelines: an agent-based approach.</p>
Clinical AND guideline AND model	<p>Comparing computer-interpretable guideline models: a case-study approach</p> <p>GLIF3: the evolution of a guideline representation format.</p> <p>The Asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines</p>
Cinical AND guideline and modeling	<p>The syntax and semantics of the PROforma guideline modeling language</p> <p>GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines</p> <p>Modeling data and knowledge in the EON guideline architecture</p> <p>The SAGE Guideline Model: achievements and overview</p> <p>Approaches for creating computer-interpretable guidelines that facilitate decision support</p> <p>Asbru: A task-specific, intention-based, and time-oriented language for representing skeletal plans</p> <p>A flexible approach to guideline modeling.</p> <p>Computer-based execution of clinical guidelines: a review</p>

	<p>Computer-interpretable clinical practice guidelines</p> <p>From guideline modeling to guideline execution: defining guideline-based decision-support services.</p> <p>Bridging the guideline implementation gap: a systematic, document-centered approach to guideline implementation</p> <p>Free and open source enabling technologies for patient-centric, guideline-based clinical decision support: a survey</p> <p>Modellgestütztes Management in Krankenhausinformationssystemen am Beispiel der Klinischen Prozesssteuerung</p> <p>A pattern-based analysis of clinical computer-interpretable guideline modeling languages</p> <p>Approaches for creating computer-interpretable guidelines that facilitate decision support</p> <p>http://books.google.de/books?hl=de&lr=lang_de lang_en&id=5rbUkY2PRwIC&oi=fnd&pg=PA115&dq=asbru+modellierung+YWzfNwFiBZGseU9k3sxOf5tzOmo#v=onepage&q=asbru%20modellierung&f=false</p>
<p>Modellierung medizinischer leitlinien</p>	<p>Klassifikation von Sprachen zur Modellierung medizinischer Behandlungspfade</p> <p>Unterstützung medizinischer Leitlinien-Von der zielorientierten Modellierung zur proaktiven Assistenz.</p> <p>Rekonstruktion eines klinischen Behandlungspfads mithilfe domänenspezifischer Erweiterungen einer Geschäftsprozessmodellierungssprache: Anwendungsfall und ...</p> <p>[PDF] Modellierung und Verifikation medizinischer Leitlinien.</p> <p>Modellierung, Planung und Ausführung Klinischer Pfade</p> <p>Simulation medizinischer Prozesse-Der Braunschweiger Krankenhaussimulator</p> <p>Prozessorientierte und standardisierte Umsetzung von DGK-Leitlinien</p> <p>SmartCare: Automatisierung klinischer Leitlinien</p> <p>Ableitung von klinischen Pfaden aus evidenzbasierten Leitlinien am Beispiel der Behandlung des Mammakarzinoms der Frau</p> <p>Konzeptuelle Modellierung im klinischen Umfeld</p> <p>Ableitung Klinischer Pfade aus Medizinischen Leitlinien</p> <p>Dissertationen (eigene und begutachtete)</p> <p>Entwicklung einer Klassenbibliothek zur Modellierung von medizinischen Prozessen</p> <p>Navigieren statt modellieren: Flexible Prozessgestaltung durch Endanwender</p>

	<p>Möglichkeiten und Grenzen der BPMN als Werkzeug zur Modellierung medizinischer Prozesse und evidenzbasierter Behandlungspfade</p> <p>Abbildung klinischer Behandlungspfade als Modelle in der Unified Modeling Language</p> <p>" Information Requisition is the Core of Guideline-Based Medical Care: Which Information is Needed for Whom?"; Betreuer/in (nen): S. Miksch, K. Kaiser; Institut für ...</p> <p>http://pub-inf.tuwien.ac.at/showentry.php?ID=141757</p>
--	---

Tabelle 9 : Erste Literaturrecherche Ergebnisse

A3

Weitere über Referenzen der Publikationen gefundene Publikationen :

Computer-based execution of clinical guidelines: a review
Towards a comprehensive treatment of repetitions, periodicity and temporal constraints in clinical guidelines
Petri nets as a formalism for comparing expressiveness of workflow-based clinical guideline languages
Declarative and procedural approaches for modelling clinical guidelines: addressing flexibility issues
Verification of temporal scheduling constraints in clinical practice guidelines
Environment-driven skeletal plan execution for the medical domain
Free and open source enabling technologies for patient-centric, guideline-based clinical decision support: a survey
Computer-interpretable clinical practice guidelines
Formal semantics of Asbru-an overview

Tabelle 10 : Referenzierte Werke

A4

```
/** Arezzo Guideline: Chronic_Cough_Guideline **/
```

```
/** 12/02/2001, 18:15:45 **/
```

```
plan :: Chronic_Cough_Guideline ;
  caption :: 'Chronic Cough Guideline' ;
  component :: Initial_assessment ;
  component :: CXR_and_initial_Rx ;
    schedule_constraint :: completed(Initial_assessment) ;
  component :: Investigations ;
    schedule_constraint :: completed(CXR_and_initial_Rx) ;
  component :: Ineligible ;
    schedule_constraint :: completed(Initial_assessment) ;
end plan .
```

```
plan :: Asthma_evaluation ;
  caption :: 'Asthma evaluation' ;
  precondition :: result_of( Cause ) includes 'Asthma' ;
  component :: Unspecified ;
end plan .
```

```
decision :: Cause ;
  choice_mode :: multiple ;
  support_mode :: symbolic ;
  candidate :: Asthma ;
```

```

    argument :: for, ( CXR_result = Normal ) ;
    recommendation :: Netsupport( Cause, Asthma ) >= 1 ;
candidate :: PNDS ;
    argument :: for, ( CXR_result = Normal ) ;
    recommendation :: Netsupport( Cause, PNDS ) >= 1 ;
candidate :: GERD ;
    argument :: for, ( CXR_result = Normal ) ;
    recommendation :: Netsupport( Cause, GERD ) >= 1 ;
candidate :: other ;
    argument :: for, ( CXR_result = other ) ;
    recommendation :: Netsupport( Cause, other ) >= 1 ;
end decision .

plan :: CXR_and_initial_Rx ;
caption :: 'CXR and initial treatment' ;
description :: 'Scheduling of CXR and initial treatment' ;
precondition :: Chronic_cough = Yes ;
component :: Scheduling_decision ;
component :: CXR_first ;
    schedule_constraint :: completed(Scheduling_decision) ;
component :: CXR_in_parallel ;
    schedule_constraint :: completed(Scheduling_decision) ;
end plan .

plan :: CXR_first ;
caption :: 'CXR first' ;
precondition :: result_of( Scheduling_decision ) = CXR_first ;
component :: Order_CXR ;
component :: CXR_report ;
    schedule_constraint :: completed(Order_CXR) ;
component :: Non_specific_initial_treatment ;
    schedule_constraint :: completed(CXR_report) ;
end plan .

plan :: CXR_in_parallel ;
caption :: 'CXR in parallel' ;
precondition :: result_of( Scheduling_decision ) = CXR_in_parallel ;
component :: Order_CXR ;
component :: CXR_report ;
    schedule_constraint :: completed(Order_CXR) ;
component :: Initial_treatment_of_chronic_cough_1 ;
end plan .

enquiry :: CXR_report ;
caption :: 'CXR report' ;
source :: CXR_result ;
    mandatory :: yes ;
end enquiry .

action :: Discontinue_ACEI ;
caption :: 'Discontinue ACEI' ;
precondition :: ACE_related = Yes ;
procedure :: 'Observe the result of discontinuing the ACE-I for 4 weeks' ;
context :: 'For patients who developed cough shortly after beginning to take an ACE-I' ;
end action .

plan :: GERD_evaluation ;
caption :: 'GERD evaluation' ;
precondition :: result_of( Cause ) includes 'GERD' ;
component :: Oesophageal_pH_monitoring ;
component :: Oesophageal_pH_monitoring_results ;
    schedule_constraint :: completed(Oesophageal_pH_monitoring) ;
component :: Oesophageal_pH_monitoring_interpretation ;
    schedule_constraint :: completed(Oesophageal_pH_monitoring_results) ;

```

end plan .

action :: Ineligible ;
caption :: 'Guideline is not appropriate' ;
precondition :: Chronic_cough = No ;
procedure :: 'This guideline for the management of chronic cough is not appropriate for this patient.' ;
context :: 'Chronic cough is defined here as cough that lasts for at least 3 weeks. ' ;
end action .

enquiry :: Initial_assessment ;
caption :: 'Initial assessment' ;
description :: 'Baseline characteristics of the patient and their cough.' ;
postcondition :: Chronic_cough = if(Cough_duration >= 3, Yes, No) ;
source :: Cough_duration ;
mandatory :: yes ;
source :: Cough_productive ;
mandatory :: yes ;
source :: Smoker ;
mandatory :: yes ;
source :: Age_group ;
mandatory :: yes ;
source :: Sex ;
mandatory :: yes ;
source :: ACE_related ;
mandatory :: yes ;
source :: Pregnant ;
mandatory :: yes ;
source :: Presumed_PNDS ;
mandatory :: yes ;
end enquiry .

plan :: Initial_treatment_of_chronic_cough_1 ;
caption :: 'Initial treatment of chronic cough' ;
component :: Discontinue_ACEI ;
component :: Non_specific_initial_treatment ;
end plan .

plan :: Investigations ;
caption :: 'Investigations' ;
component :: Cause ;
component :: PNDS_evaluation ;
schedule_constraint :: completed(Cause) ;
component :: GERD_evaluation ;
schedule_constraint :: completed(Cause) ;
component :: Asthma_evaluation ;
schedule_constraint :: completed(Cause) ;
component :: Other_investigation ;
schedule_constraint :: completed(Cause) ;
end plan .

action :: Non_specific_initial_treatment ;
caption :: 'Non-sepcific initial treatment' ;
procedure :: 'Initial treatment of the patients symptoms as appropriate.' ;
end action .

action :: Oesophageal_pH_monitoring ;
caption :: 'Oesophageal pH monitoring' ;
procedure :: '24 hour Oesophageal pH monitoring' ;
context :: '24 hour Oesophageal pH monitoring is the most diagnostically useful test for evaluating GERD as the cause of cough.' ;
end action .

decision :: Oesophageal_pH_monitoring_interpretation ;
caption :: 'Oesophageal pH monitoring interpretation' ;

```

description :: 'Conventional indices used by gastroenterologists to assess for esophagitis may be misleading normal. Until
further studies provide better guidelines, the test should be read as normal when conventional indices are within the
normal range and no suspicious reflux-induced coughs appear during the monitoring session (Grade II-2)';
choice_mode :: single ;
support_mode :: symbolic ;
candidate :: Normal ;
argument :: for, ( Conventional_indices = 'Within normal range' and Suspicious_coughs = No ) ;
recommendation :: Netsupport( Oesophageal_pH_monitoring_interpretation, Normal ) >= 1 ;
candidate :: Abnormal ;
argument :: for, ( Conventional_indices = 'Outside normal range' ) ;
argument :: for, ( Suspicious_coughs = Yes ) ;
recommendation :: Netsupport( Oesophageal_pH_monitoring_interpretation, Abnormal ) >= 1 ;
end decision .

enquiry :: Oesophageal_pH_monitoring_results ;
caption :: 'Oesophageal pH monitoring results' ;
source :: Conventional_indices ;
mandatory :: yes ;
source :: Suspicious_coughs ;
mandatory :: yes ;
end enquiry .

action :: Order_CXR ;
caption :: 'Order CXR' ;
procedure :: 'Order CXR' ;
end action .

plan :: Other_investigation ;
caption :: 'Other investigations' ;
precondition :: result_of( Cause ) includes 'Other' ;
component :: Unspecified ;
end plan .

plan :: PNDIS_evaluation ;
caption :: 'PNDIS evaluation' ;
precondition :: result_of( Cause ) includes 'PNDIS' ;
component :: Sinus_radiographs ;
end plan .

decision :: Scheduling_decision ;
caption :: 'Scheduling decision' ;
description :: 'Chest Radiographs should be ordered before any treatment is prescribed in nearly all patients with chronic
cough (Grade II-2)';
choice_mode :: single ;
support_mode :: symbolic ;
candidate :: CXR_first ;
argument :: for, ( netsupport( Scheduling_decision, CXR_in_parallel ) < 1 ) ;
recommendation :: Netsupport( Scheduling_decision, CXR_first ) >= 1 ;
candidate :: CXR_in_parallel ;
argument :: for, ( ACE_related = Yes ) ;
argument :: for, ( Age_group = Younger and Smoker = No and Presumed_PNDIS = Yes ) ;
argument :: for, ( Pregnant = Yes and Sex = Female ) ;
recommendation :: Netsupport( Scheduling_decision, CXR_in_parallel ) >= 1 ;
end decision .

action :: Sinus_radiographs ;
caption :: 'Sinus radiographs' ;
procedure :: '4 view sinus radiographs' ;
context :: 'Sinusitis may be the cause of PNDIS 30% of the time (60% if the cough is productive). Sinus CT scans are not
routinely recommended to evaluate for sinusitis as the cause of cough.' ;
end action .

action :: Unspecified ;
procedure :: 'No further guidance is specified' ;

```

```
end action .

data :: ACE_related ;
  type :: boolean ;
  true_value :: Yes ;
  false_value :: No ;
end data .

data :: Age_group ;
  type :: text ;
  range :: Younger, Older ;
end data .

data :: Chronic_cough ;
  type :: boolean ;
  true_value :: Yes ;
  false_value :: No ;
end data .

data :: Conventional_indices ;
  type :: boolean ;
  true_value :: 'Within normal range' ;
  false_value :: 'Outside normal range' ;
end data .

data :: Cough_duration ;
  type :: integer ;
  unit :: Weeks ;
end data .

data :: Cough_productive ;
  type :: boolean ;
  true_value :: Yes ;
  false_value :: No ;
end data .

data :: CXR_result ;
  type :: text ;
  range :: Normal, Other ;
end data .

data :: Pregnant ;
  type :: boolean ;
  true_value :: Yes ;
  false_value :: No ;
end data .

data :: Presumed_PNDS ;
  type :: boolean ;
  true_value :: Yes ;
  false_value :: No ;
end data .

data :: Sex ;
  type :: text ;
  range :: Male, Female ;
end data .

data :: Smoker ;
  type :: boolean ;
  true_value :: Yes ;
  false_value :: No ;
end data .
```

```

data :: Suspicious_coughs ;
  type :: boolean ;
  true_value :: Yes ;
  false_value :: No ;
end data .

```

Index

- Arden** , 3, 8, 20, 21, 22, 23, 24, 25, 27, 28, 29, 30, 63, 67, 71, 72, 86, 94, 95, 96, 98, 99, 102, 104, 105, 106, 107
- Asbru** , 3, 13, 20, 31, 32, 33, 34, 36, 40, 41, 42, 44, 45, 46, 49, 94, 95, 96, 97, 98, 99, 104, 105
- BPMN**, 4, 7, 14, 85, 86, 87, 88, 89, 95, 96, 97, 98, 100, 101, 102, 103, 108
- curly braces** , 22, 27, 28, 30, 105
- Dharma**, 20, 50, 51, 52, 53
- EEPK**, 7, 14, 88, 89, 90, 94, 96, 98, 99, 101, 102, 103
- EON** 4, 20, 50, 51, 52, 53, 54, 94, 95, 96, 97, 98, 99, 104, 105, 107
- GEM**, 20, 63, 64, 65, 66, 94, 95, 98, 104, 105
- GLARE**, 4, 20, 55, 56, 57, 61, 62, 83, 94, 95, 96, 97, 98, 99, 104, 106, 107
- GLIF**, 4, 20, 65, 67, 68, 69, 71, 72, 73, 74, 93, 94, 95, 96, 97, 98, 100, 104, 105, 107
- GUIDE**, 55, 76, 77, 95, 96, 97, 98, 104, 106, 107
- Health Level 7** , 8, 21
- Java** 12, 29, 66, 77
- klinischer Algorithmus**, 3, 20, 47, 48, 94, 95, 96, 100, 101
- klinischer Pfad**, 9, 11, 111
- Krankenhausinformationssystem**, 30, 72, 75, 77, 78, 79
- Leitlinie**, 2, 9, 11, 12, 13, 16, 22, 31, 47, 49, 50, 51, 52, 53, 54, 55, 56, 58, 59, 60, 61, 63, 64, 65, 66, 67, 68, 69, 70, 72, 73, 75, 76, 77, 78, 79, 81, 82, 84, 86, 89, 90, 92, 93, 95, 96, 99, 100, 101, 102, 103, 104, 105, 106, 107
- MLM**, 21, 22, 23, 24, 26, 27, 28, 29, 30, 96, 105
- Ontologie** , 12, 13, 50, 53, 96, 105
- Prestige**, 20, 77, 78, 79, 80, 94, 95, 96, 100
- PROforma**, 20, 82, 84, 95, 96, 97, 98, 100, 104, 106, 107
- skeletaler Plan**, 12, 13, 31, 106
- Slot**, 22, 26, 27, 94, 98
- Syntax**, 8, 20, 21, 22, 24, 27, 28, 29, 45, 46, 49, 54, 63, 67, 69, 71, 79, 86, 102
- UML** 7, 14, 73, 84, 90, 91, 92, 93, 94, 95, 96, 97, 98, 100, 101, 103, 108

Symbolverzeichnis / Abkürzungsverzeichnis

Abbildung 1 : Kernklassen des RIM mit denen alle Informationsaustauschvorgänge ermöglicht werden.....	8
Abbildung 2 : Zeitlinie der Modellierungssprachen , adaptiert nach [21].....	19
Abbildung 3 : Arden Lebenszyklus (von).....	23
Abbildung 4 : Arden Syntax Beispiel von [77]. Arden Syntax Schlüsselwörter sind dick geschrieben.....	25
Abbildung 5 : Architektur des Softwaresystems AKH Wien 2012 [54].....	30
Abbildung 6: Zeitrepräsentation von Asbru	33
Abbildung 7 : Planzustände von Asbru	34
Abbildung 8 : Intention Beispiel aus der Asbur Reference Manual [44]	36
Abbildung 9 : Beispiel von Bedingungen in Asbru aus der Asbru Reference Manual [44].....	38
Abbildung 10 : Preference Beispiel aus der Asbru 7 Reference Manual [44]	39
Abbildung 11 : Asbru Effect Beispiel aus der Asbru Reference Manual [44]	40
Abbildung 12 : XML Ausbau eines Asbru PLANes nach Reference Manual [44].....	41
Abbildung 13 : PLAN-Library aufbau nach Asbru Reference Manual [44]	42
Abbildung 14 : Beispielvisualisierung eines sequential Plans.....	43
Abbildung 15 :Beispielvisualisierung eines parallelen Plans	43
Abbildung 16 : Beispielvisualisierung eines Any-order Plans	43
Abbildung 17 : Planbeispiel von Asbru aus der Asbru Reference Manual [44].....	44
Abbildung 18 : ASBRUView Beispielhafte Darstellung der Behandlung von chronischem Husten von [80].....	45
Abbildung 19 : Terminologie Klinischer Algorithmus	48
Abbildung 20 : Sehr einfaches Beispiel eines klinischen Algorithmus von [93]	48
Abbildung 21 : Beispiel einer temporalen Abfrage in EON	51
Abbildung 22 : Graphische Darstellung einer Influenza Impfung in EON.....	53
Abbildung 23 :Architektur von GLARE. Vierecke sind Berechnungsmodule und Kreise repräsentieren Wissensbasen (Datenbanken). Der linke Teil entspricht der Auquisition von Daten, der Rechte der Ausführinheit.....	56
Abbildung 24 : GLARE Repräsentationsformalismus einer composite action.....	60

Abbildung 25 : Erfassungsmodul CG-AM. Gelbe Rauten stellen Decisions dar, Orange Dreiecke sind conclusions (Ergebnisse einer diagnostic decision), rote kreise sind composite Aktionen, blaue Kreise sind conclusions (Ergebnisse einer therapeutic DEcision) und grüne Vierecke sind Querys	61
Abbildung 26 : GEM 3 Dokumentenarchitektur [36]	64
Abbildung 27 : GEM 3 Knowledge Komponenten	64
Abbildung 28 : graphische Darstellung von chronischem Husten (aus [18])	68
Abbildung 29 :graphische Darstellung Vaccine Leitlinie (aus).....	68
Abbildung 30 : Beispielhafte Darstellung für den Medical Knowledge Layer aus der GLIF 3.5 specification Manual in UML	73
Abbildung 31 : Ausschnitt einer im NewGUIDE Tool Erstellten Leitlinie (Diagnosevorgang) ..	76
Abbildung 32 : Syntax und Semantik des Prestige Protocol Modells	79
Abbildung 33 : NICE Plan für die Behandlung einer Angina	81
Abbildung 34 : PROforma Komponentenarchitektur	82
Abbildung 35 : BPMN Sprachelemente [110]	86
Abbildung 36 : BPMN Ausschnitt für Schlaganfälle in der Notaufnahme im Charité Berlin von [55]	87
Abbildung 37 : BPMN der Abteilung für Radiologie für Schlaganfälle im Charite Berlin [55] ..	87
Abbildung 38 : BPMN Abteilung Neurologie, Transport, Labor für das Schlaganfallscenario [55]	87
Abbildung 39 : EPK Einfaches Beispiel	89
Abbildung 40 : Ablaufplanung, Einbettung und Patientendaten n.m. = nicht möglich.....	96
Abbildung 41 : Anforderungserfüllung der Modellierungsansätze adaptiert nach [63] X Anforderung erfüllt O Anforderung Teilweise erfüllt - Anforderung nicht erfüllt qu = Quellenlage unzureichend.	100

Tabellen

Tabelle 1 : Übersicht der Leitlinientypen.....	11
Tabelle 2:Ausgewählte Modellierungssprachen.....	20
Tabelle 3 : Vergleich graphischer Darstellungen von GLIF 3	68
Tabelle 4 : TOP Level Ansicht der GLIF Klassen und Attribute.....	69

Tabelle 5 : Grundelemente der Sprachen (NA : Keine Informationen verfügbar * EON und GLIF haben einen Ausführungsstatus, sind aber nicht teil des Repräsentationsmodells, sondern Teil der Ausführungseinheiten) Teilweise adaptiert nach [93]	95
Tabelle 6 : Anforderungen von Shiffman an Spracehn zur Modellierung von Leitlinien X = Anforderung erfüllt - = Anforderung nicht erfüllt	101
Tabelle 7 : Aktueller Status der Sprachen.....	107
Tabelle 8 : Recherche Sprachen Zusammenfassung	109
Tabelle 9 : Erste Literaturrecherche Ergebnisse	112
Tabelle 10 : Referenzierte Werke	112

Literaturverzeichnis

- [1] 4ed_05.dvi.
- [2] ABRU 2009 Tu Wien: Collaborative Encoding of Asbru Clinical Protocols | Information Engineering Group. <http://www.cvast.tuwien.ac.at/node/322>. Abgerufen am 11.06.2014.
- [3] Algorithmus — Leitlinien.de. <http://www.leitlinien.de/leitlinienmethodik/leitlinien-glossar/glossar/algorithmus>. Abgerufen am 08.07.2014.
- [4] AMWF Leitlinien Autor: AWMF: Aktuelle Leitlinien. <http://www.awmf.org/leitlinien/aktuelle-leitlinien.html>. Abgerufen am 14.05.2014.
- [5] Andreas Seyfang, Robert Kosara, Silvia Miksch: Asbru 7.2 Reference Manual.
- [6] Andrzejewski, D, Ledwon, P, Beck, E (05.06.2012): Möglichkeiten und Grenzen der BPMN als Werkzeug zur Modellierung medizinischer Prozesse und evidenzbasierter Behandlungspfade. <https://www.thieme-connect.com/ejournals/abstract/10.1055/s-0032-1313371>. Abgerufen am 06.02.2014.
- [7] Arden2ByteCode Arden Syntax Compiler. <http://arden2bytecode.sourceforge.net/>. Abgerufen am 11.06.2014.
- [8] ARIS: ARIS Community. <http://www.ariscommunity.com/>. Abgerufen am 08.07.2014.
- [9] ARIS Express - Free Modeling Software. <http://www.ariscommunity.com/aris-express>. Abgerufen am 13.06.2014.
- [10] asbru syntax homepage (24.09.2012): Asbru Syntax. http://www.asgaard.tuwien.ac.at/plan_representation/asbrusyntax.html. Abgerufen am 11.06.2014.
- [11] ASBRUVIEW (24.09.2012): Tools - AsbruView. <http://www.asgaard.tuwien.ac.at/tools/asbruvew.html>. Abgerufen am 11.06.2014.
- [12] ASTM (14.06.2014): ASTM E2210 - 12 Standard Specification for Guideline Elements Model version 3 (GEM III)—Document Model for Clinical Practice Guidelines. <http://www.astm.org/Standards/E2210.htm>. Abgerufen am 16.06.2014.
- [13] Author, Thomson, R (08.07.2013): OpenClinical: PRODIGY. http://www.openclinical.org/gmm_prodigy.html. Abgerufen am 04.06.2014.
- [14] ÄZQ: Leitlinien.de. <http://www.leitlinien.de/>. Abgerufen am 14.05.2014.

- [15] (05.01.2014): bflow* Toolbox - Open-Source Geschäftsprozessmodellierung. <http://www.bflow.org/>.
Abgerufen am 13.06.2014.
- [16] Biomedical Informatics Research and Development Center. <http://www.birdlab.org/research-glif.cfm>.
Abgerufen am 20.02.2014.
- [17] Bloechle: Paper_TK_Hannover_2010_09_15_blöchle.
- [18] Boxwala, AA, Peleg, M, Tu, S, Ogunyemi, O, Zeng, QT, Wang, D, Patel, VL, Greenes, RA, Shortliffe, EH (2004):
GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines.
- [19] British Medical Journal: British Medical Journal.
- [20] Bury, J., Fox, J., Sutton, D.: The PROforma guideline specification language: progress and prospects.
- [21] Christian Fuchsberger: Entwicklung einer Ausführereinheit für ASBRU Light.
- [22] Ciccarese, P, Thomson, R (08.07.2013): OpenClinical: GUIDE. http://www.openclinical.org/gmm_guide.html.
Abgerufen am 21.05.2014.
- [23] David Isern*, Antonio Moreno: Computer-based execution of clinical guidelines: A review.
- [24] DAVID R. SUTTON,BA, PHD, JOHN FOX,BSC,PHD: The Syntax and Semantics of the PROforma Guideline
Modeling Language.
- [25] de Clercq, Paul A, Blom, JA, Korsten, HH, Hasman, A (2004): Approaches for creating computer-interpretable
guidelines that facilitate decision support. *Artificial Intelligence in Medicine*, 31(1):1–27.
- [26] DIN: DIN 66001 Sinnbilder für Datenfluß- und Programmablaufpläne.
- [27] Dixon, M: GEM As a CDS Standard.
- [28] Dongwen Wang, M.Phil.1 Mor Peleg, Ph.D.2 Samson W. Tu, M.S.2 Aziz A. Boxwala, M.D., Ph.D.3 Robert A.
Greenes, M.D., Ph.D.3 Vimla L. Patel, Ph.D., D.Sc.1 Edward H. Shortliffe, M.D., Ph.D.1: Representation
Primitives, Process Models and Patient Data in Computer-Interpretable Clinical Practice Guidelines:. A
Literature Review of Guideline Representation Models.
- [29] doubleSlash Net-Business GmbH, Friedrichshafen, Net-Business, d: So nutzen Sie Visio 2007 Shapes zur
Prozessmodellierung - Business und Software Blog - Marketing, Vertrieb und Service.
<http://blog.doubleslash.de/visio-2007-shapes-zur-prozessmodellierung/>. Abgerufen am 13.06.2014.
- [30] Eisentraut, K: Abbildung klinischer Behandlungspfade als Modelle in der Unified Modeling Language.
- [31] Eklund, P, Johansson, M, Karlsson, J, Åström, R: BPMN and its Semantics for Information Management in
Emergency Care.
- [32] Elliot B. Sloane, William P. Wagner: Clinical Engineering Use of Event-Driven Process Chain (EPC) to Develop
Enterprise Models of Collaborative Information and Workflow for Re-Engineering the Operating Room.
- [33] EPC — LibreOffice Extensions. <http://extensions.libreoffice.org/extension-center/epc>. Abgerufen am
13.06.2014.
- [34] Gadatsch-IT-gestütztes Prozessmanagement im Gesundheitswesen.
- [35] (14.03.2013): GEM - Element Definitions. http://gem.med.yale.edu/Element_Definitions/element_defs.htm.
Abgerufen am 13.06.2014.
- [36] (05.03.2013): GEM - hierarchy. <http://gem.med.yale.edu/Hierarchy/hierarchy.htm>. Abgerufen am 16.06.2014.
- [37] (26.10.2012): GEM - Tools. http://gem.med.yale.edu/GEMTools/gem_ii_tools.htm. Abgerufen am 13.06.2014.
- [38] Gordon, C: Guidelines in Healthcare: the experience of the Prestige project.

- [39] Gudrun Hübner-Bloder, Dr. med. univ.: 3LGM2-basierte Modellierung von Krankenhausinformationssystemen bei der Erstellung eines IT-Rahmenkonzeptes.
- [40] Hammer, M, Champy, J (1994): Business reengineering. Die radikalkur für das unternehmen. 3. Auflage. Campus Verlag, Frankfurt.
- [41] Health Level 7: About Health Level Seven International. <http://www.hl7.org/about/index.cfm?ref=nav>. Abgerufen am 14.05.2014.
- [42] Health Level 7: Arden Syntax. <http://www.hl7.org/Special/committees/arden/index.cfm>. Abgerufen am 18.02.2014.
- [43] Hings, S: Leitlinien: der Deutschen Gesellschaft für Kardiologie – Herz- und Kreislaufforschung e.V. <http://leitlinien.dgk.org/>. Abgerufen am 14.05.2014.
- [44] HL7: Asbru Reference Manual.
- [45] HL7 Benutzergruppe Deutschland: Arden Syntax for Medical Logic Systems.
- [46] InferMed Limited: Arezzo Pathways.
- [47] internetagentur bonn köln sunzinet typo3 und redden programmierung: AWMF: Historie. <http://www.awmf.org/die-awmf/historie.html>. Abgerufen am 03.03.2014.
- [48] internetagentur bonn köln sunzinet typo3 und redden programmierung: AWMF: Leitlinien. <http://www.awmf.org/leitlinien.html>. Abgerufen am 03.03.2014.
- [49] internetagentur bonn köln sunzinet typo3 und redden programmierung (03.03.2014): AWMF: Aktuelle Leitlinien. <http://www.awmf.org/leitlinien/aktuelle-leitlinien.html>. Abgerufen am 03.03.2014.
- [50] ISO.org (20.08.2014): ISO/IEC 19505-2:2012 - Information technology -- Object Management Group Unified Modeling Language (OMG UML) -- Part 2: Superstructure. http://www.iso.org/iso/catalogue_detail.htm?csnumber=52854. Abgerufen am 20.08.2014.
- [51] Jeckle, M (20040607): Unified Modeling Language (UML) Tools. <http://www.jeckle.de/umltools.htm>. Abgerufen am 20.08.2014.
- [52] Jenders, R, Thomson, R (08.07.2013): OpenClinical: Arden Syntax. http://www.openclinical.org/gmm_ardensyntax.html. Abgerufen am 04.03.2014.
- [53] K, E, D, A, M, W, V, D (14.05.2014): gms | 53. Jahrestagung der Deutschen Gesellschaft für Medizinische Informatik, Biometrie und Epidemiologie e. V. (GMDs) | Abbildung klinischer Behandlungspfade als Modelle in der Unified Modeling Language. <http://www.egms.de/static/de/meetings/gmds2008/08gmds162.shtml>. Abgerufen am 22.08.2014.
- [54] Klaus-Peter Adlassnig, Karsten Fehre: Arden-Syntax zur Repräsentation und Verarbeitung medizinischen Wissens.
- [55] Kristian Duske¹ and Richard Müller^{2,3} and Robert Pröfeler³ and Daniel Stöhr¹: A BPMN Model of the Charite Stroke Treatment Process.
- [56] Leitlinien in der Medizin – scheitern sie an der praktischen Umsetzung? - Springer. <http://link.springer.com/article/10.1007/s001080050782>. Abgerufen am 03.03.2014.
- [57] Leitlinien von BVA und DOG. <http://augeninfo.de/leit/index.htm>. Abgerufen am 14.05.2014.
- [58] LUCILA OHNO-MACHADO, MD, PHD, JOHN H. GENNARI, PHD: The Guideline Interchange Format: A Model for Representing Guidelines.

- [59] Mark: Arezzo Pathways - Referrals - FINAL3 - 08MAY13.
- [60] Mark: BPAC Case Study - DRAFT1 - 28AUG13 AREZZO.
- [61] Mark A. Musen: EON: A Component-Based Approach to Automation of Protocol-Directed Therapy.
- [62] Martin Burwitz, Hannes Schlieter, and Werner Esswein: Modeling Clinical Pathways - Design and Application of a Domain-Specific Modeling Language.
- [63] Martin Burwitz, Hannes Schlieter, Werner Esswein: Modellgestütztes Management in Krankenhausinformationssystemen am Beispiel der Klinischen Prozesssteuerung.
- [64] (03.09.2009): Medizinische evidenzbasierte Leitlinien [www.evidence.de] Universität Witten/Herdecke. <http://www.evidence.de/Leitlinien/leitlinien.html>. Abgerufen am 14.05.2014.
- [65] Mor Peleg: GLIF : a Language for Sharing and Executing Clinical Guidelines.
- [66] Mor Peleg (25.03.2010): Guideline Representation Page. <http://mis.hevra.haifa.ac.il/~morpeleg/Intermed/GLIF.htm>. Abgerufen am 07.08.2014.
- [67] National Guideline Clearinghouse | Home. <http://www.guideline.gov/>. Abgerufen am 16.06.2014.
- [68] Nationale Versorgungsleitlinien: Nationale VersorgungsLeitlinien — Versorgungsleitlinien.de. <http://www.versorgungsleitlinien.de/themen>. Abgerufen am 14.05.2014.
- [69] NICE: Frequently Asked Questions - Evidence Search - Search Engine for Evidence in Health and Social Care. <https://www.evidence.nhs.uk/frequently-asked-questions#gen1>. Abgerufen am 04.06.2014.
- [70] NICE (15.05.2014): NICE Clinical Knowledge Summaries (CKS). <http://cks.nice.org.uk/>. Abgerufen am 04.06.2014.
- [71] Openclinical.org: SAGE.
- [72] Openclinical.org (08.07.2013): OpenClinical: Summaries of guideline representation methods. <http://www.openclinical.org/gmmsummaries.html>. Abgerufen am 13.06.2014.
- [73] Openclinical.org (08.07.2013): OpenClinical: Summaries of guideline representation methods. <http://www.openclinical.org/gmmsummaries.html>. Abgerufen am 12.06.2014.
- [74] Paolo Ciccaresea, Ezio Caffib, Lorenzo Boiocchia, Silvana Quaglinia, Mario Stefanellia: A Guideline Management System.
- [75] Paolo Terenziani: Computer Science Institute - TOSM 2010. <http://www.di.unipmn.it/en/events-en/altro-en/tosm-2010-en.html>. Abgerufen am 13.06.2014.
- [76] Paolo Terenziani and Stefania Montani and Alessio Bottrighi¹ and Gianpaolo Molino and Mauro Torchio: Applying Artificial Intelligence to Clinical Guidelines: the GLARE Approach.
- [77] Paul de Clercq: Guideline-based Decision Support in Medicine.
- [78] Paul DE CLERCQ, Katharina KAISER, and Arie HASMAN (01.01.2008): Computer-interpretable Guideline Formalisms. <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2858861/>. Abgerufen am 12.02.2014.
- [79] PD Dr. Helmut Sitte: Klinische Algorithmen.
- [80] Peleg, M (2002): Comparing Computer-interpretable Guideline Models: A Case-study Approach. Journal of the American Medical Informatics Association, 10(1):52–68.
- [81] Peter E. Friedland , Yumi Iwasaki (26.02.2012): The Concept and Implementaion of Skeletal Plans S1. <http://link.springer.com/static-content/lookinside/235/art%253A10.1007%252FBF00244995/000.png>. Abgerufen am 14.05.2014.

- [82] Peter E. Friedland , Yumi Iwasaki (26.02.2012): The Concept and Implementaion of Skeletal Plans S2.
<http://link.springer.com/static-content/lookinside/235/art%253A10.1007%252FBF00244995/001.png>.
 Abgerufen am 14.05.2014.
- [83] pubmeddev: Representing clinical guidelines ... [Stud Health Technol Inform. 2002] - PubMed - NCBI.
<http://www.ncbi.nlm.nih.gov/pubmed/15460739>. Abgerufen am 22.08.2014.
- [84] pubmeddev (22.08.2014): A UML approach to process modelli... [Stud Health Technol Inform. 2003] - PubMed - NCBI. <http://www.ncbi.nlm.nih.gov/pubmed/14664059>. Abgerufen am 22.08.2014.
- [85] Quaglini, S, Stefanelli, M, Cavallini, A, Micieli, G, Fassino, C, Mossa, C (2000): Guideline-based careflow systems. *Artificial Intelligence in Medicine*, 20(1):5–22.
- [86] Reference Information Model (RIM) Downloads. <http://www.hl7.org/implement/standards/rim.cfm>.
 Abgerufen am 27.08.2014.
- [87] Richard N. Shiffman: Using a Guideline-Centered Approach for the Design of a Clinical Decision Support System to Promote Smoking Cessation.
- [88] RICHARD N. SHIFFMAN, MD, MCIS, BRYANT T. KARRAS, MD, ABHA AGRAWAL, MD: GEM: A Proposal for a More Comprehensive Guideline Document Model Using XML.
- [89] S3-Leitlinie. <http://www.dgbs.de/s3-leitlinie.html>. Abgerufen am 14.05.2014.
- [90] Samson W. Tu: The EON Guideline Modeling System.
- [91] Samson W. Tu: Using Ontologies in Clinical Decision Support.
- [92] Samson W. Tu, M.S., Mark A. Musen, Ph.D.: The EON Model of Intervention Protocols and Guidelines.
- [93] Sarshar, K, Loos, P (2004): Klassifikation von Sprachen zur Modellierung medizinischer Behandlungspfade. *Modellierung betrieblicher Informationssysteme-MobIS*:43–59.
- [94] sascha mueller: Microsoft Word - Dissertation-SM-final.doc.
- [95] Seyfang, A, Miksch, S, Votruba, P, Rosenbrand, K, Wittenberg, J, Croonenborg, J von, Reif, W, Balsler, M, Schmitt, J, van der Weide, T (2004): Specification of formats of intermediate, asbru and kiv representations. EU Project Procure.
- [96] Shiffman, R, Thomson, R (08.07.2013): OpenClinical: GEM: the Guideline Elements Model.
http://www.openclinical.org/gmm_gem.html. Abgerufen am 13.06.2014.
- [97] Silvana Quaglini: (New-) GUIDE.
- [98] Silvia Miksch 1) Yuval Shahar 2) Peter Johnson 2): A TASK-SPECIFIC, INTENTION-BASED, AND TIME-ORIENTED LANGUAGE FOR REPRESENTING SKELETAL PLANS.
- [99] Sutton, DR, Taylor, P, Earle, K (2006): Evaluation of PROforma as a language for implementing medical guidelines in a practical context. *BMC Medical Informatics and Decision Making*, 6(1):20.
- [100] T. Knappe, L. Hederman, V. P. Wade, M. Gargan, C. Harris, Y. Rahman: A UML Approach to Process Modelling of Clinical Practice Guidelines for Enactment.
- [101] Terenziani, P: GLARE (GuideLine Acquisition Representation and Execution).
- [102] Thomson, R: Telematics for Clinical Guidelines: A Conceptual Modelling Approach.
- [103] Tina Küttner, Norbert Roeder: Definition Klinischer Behandlungspfade.
- [104] TU Braunschweig (04.05.1999): Ziele der UML. <http://www.umi.cs.tu-bs.de/sep/uml/sld009.htm>.
 Abgerufen am 20.08.2014.

- [105] Tu Wien (24.09.2012): The ASGAARD Project ::PLAN.
http://www.asgaard.tuwien.ac.at/plan_representation/protocols.html. Abgerufen am 02.07.2014.
- [106] UniversitätsKlinikum Heidelberg: HELEN 3 Projektseite. <http://www.klinikum.uni-heidelberg.de/index.php?id=8914>. Abgerufen am 04.03.2014.
- [107] Votruba, P (18.07.2006): DELTA/A. <http://ieg.ifs.tuwien.ac.at/projects/delta/>. Abgerufen am 11.06.2014.
- [108] Website DGK: Leitlinien — Arztbibliothek. <http://www.arztbibliothek.de/leitlinien>. Abgerufen am 14.05.2014.
- [109] What is athenahealth? <https://www.youtube.com/watch?v=X1EyYOCZi6o>. Abgerufen am 12.06.2014.
- [110] Wikipedia (07.05.2014): Business Process Model and Notation.
<http://de.wikipedia.org/w/index.php?oldid=130175685>. Abgerufen am 22.05.2014.
- [111] Wikipedia (08.06.2014): Qualitätsmanagementnorm.
<http://de.wikipedia.org/w/index.php?oldid=131128079>. Abgerufen am 13.06.2014.
- [112] Yuval Shahar¹, Silvia Miksch², and Peter Johnson³): The Asgaard Project: A Task-Specific Framework for the Application and Critiquing of Time-Oriented Clinical Guidelines.
- [113] ZaeFQ: Präsentation, Disseminierung, Implementierung.

Selbständigkeitserklärung gem. § 21 Absatz 5 MPO

Hiermit versichere ich,, dass ich die vorliegende
Masterarbeit mit dem Titel

.....
.....

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

.....

Ort, Datum

.....

Unterschrift