Semantic Web Technologies in the Quest for Compatible Distributed Health Records

Roland Hedayat
Abstract

Semantic Web Technologies in the Quest for Compatible Distributed Health Records

Roland Hedayat

There is a proliferation of patient bound Electronic Health Record (EHR) data in systems that are incompatible - challenging the goal of granting authorized access to the accumulated medical history of a patient, whenever requested, and whichever the source, in order to secure a safe treatment.

A common semantic representation is a prerequisite for validating the semantics of one EHR system against another. Therefore, assessing the semantic compatibility between systems implies having a formal method for extracting their semantics, and for validating the consistency of their combined semantics. A guiding hypothesis is that Semantic Web Technologies and Ontology Web Language (OWL) are potential bridging technologies between the EHRs and medical terminologies, and can be used to represent the combined semantics of the systems to be integrated. Furthermore, that automatic reasoners can perform semantic validation of the combined subsystems.

Some experimental steps in this direction are taken, preceded by a discussion on Medical Terminologies, Ontologies, EHR-systems and their interrelationships, and a summary overview of Description Logics, the Semantic Web and the Web Ontology Language, OWL.

The OpenEHR reference model is transformed from an XML-schema representation to OWL, and a couple of archetypes are transformed into OWL in a manual procedure. Subsequently, validation runs with a formal reasoner on the transformed results were performed, demonstrating the feasibility of the process.

The problems of EHR semantic interoperability are complex. Awareness of the necessity of applying formal semantic methods when dealing with inherently semantic problems will catalyze the process of solving them.
# Table of Contents

1. Swedish Summary ........................................................................................................... 5

2. Introduction .................................................................................................................. 7
   2.1. Background and Motivation .................................................................................... 7
   2.2. Problem .................................................................................................................. 8
   2.3. Working Hypothesis and Goal ............................................................................... 12
   2.4. Delimitation .......................................................................................................... 13
   2.5. Method .................................................................................................................. 14
   2.6. Report Outline ..................................................................................................... 15

3. Medical Knowledge and the EHR .............................................................................. 17
   3.1. Recording Medical Knowledge .......................................................................... 17
   3.2. The Electronic Health Record ............................................................................. 19
   3.3. Linking the Health Record to Medical Knowledge ........................................... 21

4. Theoretical Underpinnings ............................................................................................ 23
   4.1. Formal Methods, Models and Modelling ......................................................... 23
   4.2. Description Logics ............................................................................................... 24
   4.3. The Semantic Web ............................................................................................... 26
   4.4. OWL — Bringing DL to the Semantic Web ..................................................... 27
   4.5. Model Driven Architecture .................................................................................. 27

5. Related Work .................................................................................................................. 31
   5.1. Mapping XML to OWL ....................................................................................... 31
   5.2. Mapping the EHR to OWL .................................................................................. 31

6. Practical and Experimental Work .................................................................................. 33
   6.1. Open Source Tool Set .......................................................................................... 33
   6.2. Procedure Outline ............................................................................................... 33
   6.3. Mapping of the OpenEHR RM to OWL ............................................................. 34
   6.4. Conversion of Selected Archetypes to OWL .................................................... 38
   6.5. Some issues ......................................................................................................... 40
   6.6. Consistency and Satisfiability Checking ............................................................. 42

7. Discussion ....................................................................................................................... 47
   7.1. Results and Observations .................................................................................... 47
   7.2. EHR Terminology Binding — the Code Binding Interface ............................... 47
   7.3. ADL versus OWL ............................................................................................... 48
   7.4. Ontology and Ontology Alignment Issues ....................................................... 49

8. Future Work .................................................................................................................... 51
   8.1. Completing the Archetype Ontologising process ............................................. 51
   8.2. Formalisation of the Semantic Distillery ......................................................... 51
   8.3. Work on the CBI Concept ............................................................................... 51
   8.4. Optimising Reasoner Performance ..................................................................... 51
   8.5. The EHR from an MDA Perspective ................................................................. 51

9. Conclusion ....................................................................................................................... 53

Glossary ............................................................................................................................... 55

Bibliography ........................................................................................................................ 59

A. Open Source Tool Set .................................................................................................... 63
   1. XML Transformation Tools and Libraries ......................................................... 63
   2. OpenEHR Related Tools and Libraries ............................................................... 63
   3. Semantic Web Related Tools and Libraries ........................................................ 64
   4. Program Editor and Version Control System ..................................................... 64
   5. Publishing Tools and Libraries .......................................................................... 64

B. Lab Artefacts .................................................................................................................... 65
Acknowledgements

Special thanks to my supervisor, Daniel Karlsson for all the interesting discussions. Also thanks to prof. Hans Åhlfeldt for accepting to "host" my thesis work at IMT, and to Erik Sundvall for encouraging me to approach IMT in the first place.

At Uppsala I have good memories of many persons who have contributed with a friendly attitude coupled with solid professionalism during the first phase of this journey. Special thanks to Roland Bol, Bengt Sandblad and Pierre Flener.

And to E who continuously gave me her support.
1. Swedish Summary

Av historiska skäl registreras, bearbetas och lagras patientbunden information i ett stort antal inkompatibla patientjournalsystem. Denna situation utgör ett hinder för etablering av en viktig grundbult i säkerställandet av en högkvalitativ sjukvård som utvecklas i paritet med disciplinens kunskapsnivå: Skapandet av en "virtuell" journal som möjliggör tillgång till patientens totala medicinska historik när som helst, och från alla avseende patienten tillgängliga källor.

Vi är konfronterade med en teknologiparadox: Än en sidan har stora investeringar gjorts i informations- och kommunikationsteknologi inom sjukvården för att dra nytta av de potentiella fördelarna denna tillför, å andra sidan är dessa fördelar till stor del utom räckvidd på grund av bristande interoperabilitet mellan sjukvårdens fragmenterade patientrelaterade informationssystem.

Det finns goda skäl att anta att sjukvårdens system kommer att förblū lokaliserade och distribuerade: Sjukvårdens organisations- och behandlingsmässiga mångfald, ett ökande antal privata aktörer, specialistenheters divergerande tekniska behov, flexibiliteten i teknologisk mångfald mm.

Interoperabilitet mellan distribuerade patientjournalsystem kräver automatiserat utbyte av journalfragment — logistiskt självständiga patientbundna informationsentiteter. Ett sådant utbyte förutsätter i sin tur att avsändare och mottagare måste "förstå" de utbytta meddelandena på samma sätt — de måste vara semantiskt interoperabla (Figure 2, “Sharing of Crucial Components for Semantic Interoperability”).

Problemet med bristande semantisk interoperabilitet mellan distribuerade system är av generell natur. Själva Internet kan betraktas som ett stort distribuerat system, och den övervägande delen av informationen ligger utspritt på system som inte är semantiskt interoperabla eftersom standard internetteteknologi inte alls exponerar datans underliggande semantik, eller i bästa fall i en alldeles för begränsad grad. Trots att Internet har gjort en enorm mängd information tillgänglig för oss, så bekommer det fortfarande till största del människor att sälja, sammanställa och vidareförädla denna information, vilket exponerar oss för problemet med "informationsöverbelastning".


En grundläggande tes för detta arbete är följande:

For att kunna uppskatta och föra resonemang om graden av semantisk interoperabilitet mellan olika system måste det finnas en formell metod för att extrahera semantiken frå dem, och därefter en formell metod för att kunna jämföra den kombinerade semantiken från dessa system.

En allmän uppfattning är att lösningen på problemet med semantisk interoperabilitet mellan medicinska journalsystem består i sammankopplingen av dessa med medicinska terminologisystem. Detta är en nödvändig men ej tillräcklig förutsättning — det finns många sätt som två system kan vara inkompatibla på, även om båda är bundna till samma terminologisystem. Dessutom leder introduktionen av terminologisystem till uppkomsten av ett nytt problem: "the boundary problem" — hur exakt dras gränsen mellan journalsystemens semantik och terminologisystemens semantik? (Section 3.3.2, “The Interface between EHRs and Terminology Systems”). Flera medicinska terminologisystem, bl.a. SNOMED CT kan emellertid uttryckas i OWL. Tidigare arbeten har gått ut på att konvertera journalsystemens meta-data (schema) till OWL. Detta har bidragit till en arbetshypotes för detta arbete:

En vägledande hypotes för etablerandet av formella semantiska metoder är att Semantisk Webbteknologi och Ontology Web Language (OWL) kan vara potentiella brobyggarteknologier mellan pa-
tientrelaterade informationssystem, samt mellan dessa och medicinska terminologisystem. Vidare att OWL kan användas för att representera den kombinerade semantiken i de system som skall integreras, och att *automatiska inferensmotorer* (programkomponenter som utför logisk analys på modeller beskrivna i "Description Logics") kan användas för semantisk validering av ingående subsystem.

Detta arbetes mål är att ta några inledande steg för att testa denna hypotes genom ett experimentellt tillvägagångssätt. Detta förutgår av en översiktlig genomgång av kopplingen mellan medicinsk kunskap och patientbundna data samt av relevant teori, speciellt "Description Logics", och dess relation såväl till semantisk webbteknologi som till journalsystem och terminologisystem. Tidigare utförda arbete som har inspirerat eller bidragit till detta arbete berörs i ett eget avsnitt. Därmed är det begreppsmässiga ramverket för det experimentella arbetet och efterföljande diskussion etablerat.

Det praktiska arbetet syftar till att stödja arbetshypotesen genom att konvertera delar av ett journalsystems "schema" till OWL. Syftet är att skapa en mer semantiskt "ren" representation av informationen, vilket är en förutsättning för att kunna validera ett journalsystems semantik mot ett annat, och för att kunna lösa ett konsistensproblem när journalsystem skall bindas till terminologisystem.

Arbetet begränsas till operationer på modeller från OpenEHR. OpenEHRs referensmodell (RM) transformeras från deras XML-schema representation till OWL genom en XSLT-transformation (cf. Section 6.3.3, "Transformation and Validation Tools"). Detta beskrivs i Section 6.3.4, "Transformation Outline". Vidare transformeras ett par arketyper till OWL. Detta görs i en manuell process, trots att ett tidigare arbete har visat automatiske procedurer för detta (Section 5.2.2, “Mapping Archetypes to OWL”). Motiveringen för detta samt beskrivningen för proceduren i detta arbete görs i Section 6.4, "Conversion of Selected Archetypes to OWL". Den manuella transformationen visade på några underliggande problem i arketyperna som måste lösas innan en automatisk procedur kan genomföras, men metodens görbarhet demonstreras.

I ett efterföljande steg görs experimentellt validering med en formell "reasoner" (inferensmotor) på det transformerede resultatet. Dessa valideringar gav resultat som förväntat och därmed är ett "proof of concept" för proceduren etablerad. Baserat på dessa resultat föreslås framtida arbete i utstakad riktning.


Lösningarna på dessa problem lär hittas stegvis och genom fokuserad forskning. En katalysator för denna process torde vara att beakta nödvändigheten av applicerandet av formella semantiska metoder när man skall handhas med problem som till sin essens handlar om semantik.
2. Introduction

The question of whether a computer can think is no more interesting than the question of whether a submarine can swim (Edsger W. Dijkstra)

2.1. Background and Motivation

During the past couple of decades, there has been a rapid proliferation of Electronic Health Record (EHR) systems, both in the primary health care, and in the clinics. During the same period of time the Internet has become omnipresent, and we rely on it for accessing and sharing information worldwide. As a natural consequence of this development, it is expected by patients as well as by health care personnel that EHR-fragments (cf. Section 3.2, “The Electronic Health Record”) can be communicated in a secure, reliable and efficient way across health care units and system boundaries in such a way that a comprehensive view of a patient’s medical history can be presented to relevant health care providers anywhere and at any time.

2.1.1. The Technology Paradox

Unfortunately, the vision is still far from the reality. The first generations of systems were constructed without this functionality in mind. Also, since the discipline of Medical Informatics is still young, at least measured with a time scale appropriate for the processes of standardisation, there have been no enforcing standards or supporting guidelines in place to ensure such a development. The very first generation of systems mimicked the free text format of the paper based health record, and even following generations of systems lacked the necessary level of formality to enable machine to machine transfer and interpretation of EHRs.

We are confronted with a technology paradox: On one side huge investments have been made in deployments of ICT in health care in order to reap the benefits from this technology, and on the other side, the potential benefits remain largely unexploited due to the lack of interoperability among the fragmented EHRs. Telephone, fax and surface mail remain important, if not even the main channels for the communication of health record data.

2.1.2. The Big Projects Solution

Many attempts to work around the problem with large scale, centralised — "one size fits all" strategies have been made, but they tend to make the situation even more complicated and confusing:

Huge IT projects tend to fail. The overall rule for huge IT projects is that they are vastly more expensive and time consuming than projected, and most of them fail.

This phenomenon has been eloquently analysed and accounted for in the timeless classic, “The Mythical Man Month” [BROOKS95]. The first edition of this book appeared as early as 1975, but the same errors keep being repeated at largely the same pace. According to the "Chaos Manifesto", an October 15, 2009 report from The Standish Group[^1], in a (2008) sample of large IT-projects surveyed, only 32 percent delivered according to requirements, budget and time frame.

Inflexibility of Large Monolithic Systems. The ambition to create big monolithic systems incorporating a vast spectrum of needs creates inflexible solutions that are hard to maintain and therefore not “future-proof”.

Acommodating External Systems Difficult. There will always remain actors outside of the centralised system (private actors, highly specialised units and labs, other interested parties outside of the organisation etc) that will create EHR-fragments, or need access to the EHR.

[^1]: http://www.standishgroup.com
2.1.3. Distributed EHRs Require Semantic Interoperability

Therefore, more flexible and decentralised solutions are needed, and fortunately this is not science fiction. With the advances in communications and distributed systems technologies, modern and flexible system architectures consisting of loosely coupled, heterogeneous subcomponents are emerging.

Assuming that distributed architectures is the realistic way of integrating the many instances of EHR systems, communication of fragments of EHR across system boundaries become inevitable. But for this to be possible, the systems have to be interoperable in the sense that not only do they have to be able to send and receive EHR fragments, from each other, they also have to “understand” them, at least to a level where typical operations such as queries and aggregation is possible at the machine level. In general, so is not the case today. This is what semantic interoperability is all about, the lack of which remains the main barrier for realizing the vision of a Virtual EHR, where relevant EHR fragments located at any registered health care unit are assembled "on the fly” and presented in a unified, coherent form. We will delve further into some of the remaining technical/informatical problems to overcome this problem.

2.2. Problem

Why is it so hard to exchange health records between health care units in a reliable and consistent way? In the age of Internet we are used to accessing and sharing instances of many types of data over the network — often complex and heterogeneous, such as multimedia. Why can't we do the same with the EHRs? What does it take to make the EHR exchangeable?

In a typical "web usage" scenario, the user interacts with a set of web servers via her web client, resulting in data entities of various types being exchanged between the server and client. The web browser has a sufficient understanding of these types to be able to "render” it, a process of making the information entities human intelligible by displaying text, pictures, playing video or sound streams and so forth. The task of making sense of — attaching semantics to the assembled “information entities” is entirely up to the human user, and there is nothing preventing the information created and presented from being meaningless, inconsistent or ambiguous.

For our vision of a distributed EHR system the requirements are at a higher ambition level. In order to enable computers to exchange and aggregate EHR data from different sources in a distributed environment, and furthermore to process it for purposes such as queries and data structure navigation as well as for analysis and decision support, several additional requirements must be met. I will elaborate on these in turn.

2.2.1. Syntactical Interoperability in a Complex Environment

EHR data is typically complex, consisting of large collections of data entities of many types, with many interrelationships, and even many types of interrelationships. The challenge is to represent this data in a way that both permits all the facets of it to be captured, and also in a flexible way, such that changes over time are handled with a minimum of effort, without loss of information, and ensuring backwards compatibility.

When EHR-fragments are transmitted "over the line" from one system to another, the data structures, typically complex, are transformed into linear sequences of symbols, they are "serialised”. The task of the receiver is to "deserialise” the data stream, in order to build up a structure which is equivalent to that of the sender. For this to be feasible, the parties must adhere to some protocol, and also have access to the same schema. In this case, the role of the protocol would be to set the rules for how the data structure is serialised/deserialised, while the schema contains the necessary information to build the various entities from the serialised stream. If the parties in this process are able to recreate each others data structures with equivalent information, and to the same level of granularity, then we may say that the systems are "structurally compatible" with each other.

Structural compatibility in itself is no strong requirement, since it is a prerequisite for any meaningful exchange of data between applications. But to ensure this property together with the complexity, diversity, changeability and flexibility requirements above make it more of a challenge. The two previous
requirements combined have therefore been subject of research in the field of medical informatics for
the last couple of decades (cf. Section 3.2, “The Electronic Health Record”).

### 2.2.2. Coping with Semantics

#### The Human Way.

However, the problem of semantics — of a common interpretation conveying the meaning of the data remains to be addressed. The human communications process is illustrated (in a very simplified form) in Figure 1, “Amazing Robustness of Human Communication”. When humans exchange text or oral messages, both sender and receiver activate a "syntactic/semantic processor" in their brains \( \text{NLP}_1 \) and \( \text{NLP}_2 \) and a "dictionary" or Vocabulary, \( \text{V}_1 \) and \( \text{V}_2 \). The semantic processor and vocabulary of each party are subsets of some conceived total counterpart \( \text{NLP} \) resp. \( \text{V} \), residing in the "cultural context". The exact "program" for the processor to use might be negotiated in an initial negotiation with phrases like "Do you speak English..." but in most cases, the processors are implicit, the communicating parties simply assume that the other side possesses a sufficiently equal "language processor" and vocabulary. During message exchange, the sender "looks up" "entities of meaning" which in itself is a complex process that has intrigued linguists and philosophers\(^2\), \( \text{V}_1 \) in the brain's memory. Now she uses her "language processor" \( \text{NLP}_1 \) to encode these to terms which in turn are used in the last stage, which is building syntactically correct phrases reflecting the combined meaning of the terms. The receiving process is the reverse, the phrases are decomposed to find the syntactical structure and the terms, and from this information, the meaning of the terms is "looked up" in the brain of the receiver. The meaning of the set of terms is then matched with their position in the syntactical structure, and from this information, the receiver builds up an interpretation of the meaning of the message as a whole. The two parties share the intersection of their vocabularies, and language processors (the shaded areas in the figure), and are at first not aware of what they share. Convergence upon usage of the common subsets is a learning process during the conversation.

![Figure 1. Amazing Robustness of Human Communication](image)

There is a degree of fuzziness in this process that leaves room for misunderstandings and errors. The sender and receiver may not share exactly the same vocabulary (set of words in the language), and may not agree totally upon what each word means. In spite of this, human to human communications is astonishingly efficient and a prerequisite for human civilisation. The way humans distill the essence out of fuzzy information has been subject of much research, and inspired a new branch of mathematical logic, fuzzy logic\(^3\) [ZADEH04].

#### Computers and Semantics.

When analysing the problems involved in exchanging EHR-fragments between computers, the analogy with human to human communications is useful. But on one hand, we have not yet reached the level of sophistication necessary for coping with vagueness and ambiguity using fuzzy logics or related techniques upon exchange of EHR-data. On the other hand, within the

---

\(^2\) [http://conceptualgraphs.org/]

\(^3\) [http://www.csiss.org/classics/content/68]
field of medicine, where human life is at stake, we cannot accept even the astonishingly low "misunderstanding rate" which is inherent in natural language processing between humans. We will instead try to eliminate ambiguity and vagueness through the usage of explicit sharing and formal methods (cf. Section 4, “Theoretical Underpinnings”). The idea is to replace those parts that are implicit in human to human communication (vocabulary, grammar, definition of terms and concepts) with explicit and machine interpretable references to such knowledge in the machine to machine communication. In figure Figure 2, “Sharing of Crucial Components for Semantic Interoperability”, both the sending and receiving machines $C_1$ and $C_2$ share the same copy of the terminology, and have local copies of the Parser/Serialiser ($P/S$) as well as of the Semantic Processor ($SPr$).

![Diagram of Explicit Sharing of Terminology, Syntactic and Semantic Processors for Semantic Interoperability between Machines](image)

Figure 2. Explicit Sharing of Terminology, Syntactic and Semantic Processors for Semantic Interoperability between Machines

These local copies are either identical to, or proven equivalent with the reference implementations, residing in the "Explicit Common Context" ellipsis. The requirements for successful machine to machine processing of knowledge are strict, and everything is explicit.

### 2.2.3. A General Problem

The problem we are facing is of more general nature, and not confined to the realm of health care systems only. The Internet and the Web have made vast quantities of information available "at our fingertips". But for the major part of this data, the semantics of it has not been made explicit. Therefore, humans are the only arbiters of the relevance and priority of the served information. They are thereby exposed to the phenomenon of "information overload" — a state of saturation when processing and filtering overwhelming quantities of information when performing the semantic processing manually.

The creators of the World Wide Web\(^4\) were aware of these problems. In the late nineties Tim Berners-Lee [BERNERSLEE01] and his team took the initial steps to realize a vision dubbed the Semantic Web. The original vision of the Semantic Web, constituted a web of data with semantic annotations, provided by a new language specifically designed for describing meta-data on the web, Resource Description Framework, (RDF).

In parallel, several research activities within the field of computer based knowledge representation had led to convergence toward an entirely new discipline of Logic, Description Logics (DL) (cf. Section 4.2, “Description Logics”) — a formal logics method for describing the semantics of the entities and concepts within a domain of interest. The driving force which led to the rapid advance of DL was the capabilities of applying automated formal reasoning techniques about the entities of a domain described by DL. Automated reasoning is a powerful capability. It may uncover “hidden” facts that are consequences of stated facts, and as a special case, inconsistencies within the described model. Automated reasoning is made possible due the rigid theoretical foundations of this sub discipline of Mathematical Logic. Domains described by DL have (at least within computer science) become synonyms with ontologies (cf. Section 3.1.2, “Ontologies”)

Already in the embryonic phase of the Semantic Web it was realized that RDF was not enough to realize the vision. A more complete formalism was needed, and this led (with an intermediate step,

\(^4\) http://en.wikipedia.org/wiki/Tim_Berners-Lee
RDFS) to the conception of a new language, Web Ontology Language (abbreviated OWL), a cross fertilisation between Description Logics and the technologies of the Web.

The knowledge base of medicine is very large in itself, and combined with the other biomedical sciences it is huge. In order to make it more manageable, various efforts have been undertaken to organise it by building controlled vocabularies (cf. Section 3.1, “Recording Medical Knowledge”). Some of these have evolved into full blown Description Logics based ontologies. In fact the demand from large biomedical terminology systems has served as an accelerator of the development of DL. But the influence is clearly two way, advances in the field of Description Logics also has had a catalysing effect on the development of biomedical terminology systems and ontologies such as SNOMED CT and OpenGALEN5.

Thus, the principles for building modern medical ontologies and the Semantic Web share common ground, both being based in Description Logics. The results from within Description Logics and the Semantic Web may have a great impact also on the problem of semantic interoperability for medical applications. This is the track I will follow in the present work.

2.2.4. Problem Statement

One of the stated purposes of medical terminology systems and ontologies is to link them to the EHR. The aim of such a linkage is to attribute formal and sharable semantics to the entities and data structures in the EHR. The linkage is called a terminology binding (cf. Section 3.3.2, “The Interface between EHRs and Terminology Systems”).

Let us consider the simplest case of potential interoperability problem, regarding two Electronic Health Care Record Systems, EHR1 and EHR2, and one common formal terminology system, T. The systems EHR1 and EHR2 are supposedly built with different implementation techniques (programming language, internal data structures etc.), but both have an established terminology binding to T.

We want to make assessments about the semantic interoperability of the involved systems, (with focus on semantic, deliberately leaving out all other aspects of interoperability) by getting answers to questions like:

1. Is the terminology binding of EHR1 to T "correct"?
2. Is the terminology binding of EHR2 to T "correct" and "complete"?
3. If the answer to 2 is 'no', is there any binding EHR2 to T such that it is "correct" and "complete"?
4. Are the two systems EHR1 and EHR2 "semantically compatible"?

The traditional way to cope with this type of problem is to establish a terminology binding for each system EHR1 and EHR2, and to establish a mapping between the systems, such that data elements sent from EHR1 are transformed into corresponding data elements of EHR2 upon receipt and vice versa. The next step is to establish a set of test cases aiming at answering the questions above, and to verify that the combined system behaves as intended. But there are (at least) two serious problems with this approach:

• The method does not scale, because the number of necessary mappings grow quadratically (n * (n-1)/2) with the number of systems to integrate

• “Testing shows the presence, not the absence of bugs”. This quote from E. Dijkstra6 pertinently poses the problem. With large EHR systems and large ontology/terminology systems it is impossible to ensure consistency and completeness with testing only

On a larger scale than that of just few systems combined, the trial and error mapping method will not work. We will neither have the resources to create all the peer to peer mappings, and even less so, to maintain them as the subsystems evolve, nor will we be able to safely assess the correctness of the

5 www.opengalen.org
(ever changing) combined set of mappings. Thus we need some sort of "semantic distillery" to enable us to establish a common semantic representation among the set of subsystems.

This leads to the following problem statement:

In order to validate the semantic compatibility between systems, there must be a formal method for extracting the semantics from them, and a formal method for validating the consistency of the combined semantics thus extracted.

An overview of this process is shown in Figure 3, “Semantic Distillation and Reasoning”. The EHR-systems encapsulate the entire semantics of their health record in their data structures and schemata, but in a form not suitable for reasoning. The “semantic distillation” is a typical case of an abstraction process. By systematically filtering out those parts of the model (or the object of study) that do not carry “relevant semantics” we are enabled to focus on and reason about the essentials that are left. The hope and hypothesis (cf. Section 2.3, “Working Hypothesis and Goal”) is that even machines will be able to reason over the abstract models. Automated formal reasoning is treated in Section 4.2, “Description Logics” and in Section 4.4, “OWL — Bringing DL to the Semantic Web”.

![Semantic Distillation and Reasoning](image)

Figure 3. Enabling Formal Reasoning by Semantic Distillation

This is probably a hard task to realize to a full extent, and opens up for difficult follow-up questions, some of which will be treated in Section 7, “Discussion” and in Section 8, “Future Work”. This problem statement will nevertheless guide the work, as described in the next sections.

2.3. Working Hypothesis and Goal

One of the main aims of the present work is to contribute to the investigation of the suitability of Description Logics and Semantic Web technologies as a vehicle for bridging the gap between the EHR and medical knowledge management systems. Therefore, the Web Ontology Language (OWL) will be tested for usability as this common target language.

2.3.1. Justification of Technology Choice

The usage of Semantic Web technologies and OWL for our purpose is justified for the following reasons:

- **"Web Awareness" Important.** Some of the properties of OWL are particularly suited for the problem at hand, such as being designed for the Web, which fits well into the vision of a "virtual" EHR system, composed of instances distributed over the Web.

- **OWL Representation already in Several Terminology Systems.** The conversion of several biomedical terminology systems to OWL is already done, one prominent example of which is SNOMED CT. The SNOMED CT distribution comes with a program that performs the transformation into OWL.

- **A Description Logics Language of Increasing Acceptance.** There is considerable activity around the Semantic Web and OWL in the W3C and in science, notably within the Biomedical
2.3.2. Working Hypothesis

The discussion so far calls for formulating a hypothesis which will guide the work:

Semantic Web technologies/OWL are potential bridging technologies between the EHRs and the terminology system(s) to be bound to it. OWL can be used for representing the combined semantics of the systems to be integrated, and automatic reasoners can be used for semantic validation of the sub-systems against each other, and against the terminology system.

This hypothesis relies on the following assumptions:

- **The EHR systems are expressible in OWL without (relevant) information loss.** This assumption is backed by related work and Section 5.2, “Mapping the EHR to OWL” that strongly suggests that this is the case at least for OpenEHR. The claim will be further justified by the observations in the practical work as shown below.

- **The Terminology system is expressible in OWL without (relevant) information loss.** This is "de facto" proven in the case of SNOMED CT, where the distribution contains a tool to convert from SNOMED CT to OWL.

- **There is a method for binding the Terminology to the EHR system with OWL as bridging technology.** Related work have given examples of such a method, and some issues related to this will be further discussed in Section 7.2, “EHR Terminology Binding — the Code Binding Interface”.

2.3.3. Goal

The work will consist of transforming parts of the OpenEHR's information structures into the Ontology Web Language, OWL. The aim is to create a more pure semantic representation of the information in the EHR. This is a prerequisite for validating the semantics of one EHR system against another, and for resolving the "boundary problem" (cf. Section 3.3.2, “The Interface between EHRs and Terminology Systems”) when establishing terminology bindings with established terminology systems.

I will then do some experimental validations with a formal reasoner on the transformed result, followed by a discussion of issues and results of this process.

Based on results and observations from the previous steps, I will discuss the viability and merits of the proposed path, and how the result of this work can be linked with other related work, and point out remaining problems and work ahead.

2.4. Delimitation

As previously stated, there are many different EHR systems and several medical terminology and ontology systems based on Description Logics. The problem involves various domains of theory, such as Logic, and Model Theory. The following delimitations will be made.

2.4.1. Restriction of Scope

**One EHR.** The work will be limited to the transformation of EHRs based on the OpenEHR specifications.

OpenEHR is well suited as the first candidate for transformation for the following reasons:

- The OpenEHR specifications are open, and there is a community to discuss issues with

---

7 [http://www.openehr.org/](http://www.openehr.org/)
• The two level modelling approach based on Archetypes (cf. Section 3.2.1, “The OpenEHR Foundation”) is gaining increasing interest, specially in European health care organisations, partly because of their recent acceptance by the CEN/ISO standards bodies (cf. Section 3.2.3, “Standards Supporting the Archetype Concept”).

• Related work has been done to analyse the archetype concept from a description logics viewpoint. The OpenEHR Reference Model has been translated into OWL, and tools have been built to automatically transform Archetype definitions to OWL (cf. Section 5.2, “Mapping the EHR to OWL”).

Terminology Binding Discussed Only. It is beyond the scope of this work to realize the next natural step after having transformed the EHR-models, i.e. establishing a terminology binding in practice. Instead this will be discussed using SNOMED CT\(^8\) for examples, for the following reasons:

• There is a tool for converting SNOMED CT to OWL that comes with the standard distribution, so there exist already a target representation of the terminology.

• SNOMED CT is gaining increasing acceptance (cf. Section 3.1.3, “SNOMED CT”) with commitment to it from various countries, among them, Sweden, Denmark, UK.

Model Driven Architecture Discussed Only. Although the case is made for looking deeper into the usage of Model Driven Architecture (MDA) in Section 4.5, “Model Driven Architecture”, it was deemed as beyond the scope of the present work to actually use the methods and principles of this discipline in the practical part (cf. Section 6, “Practical and Experimental Work”). An overview of the MDA conceptual framework is nevertheless included here because it is of great help for the understanding, and also relevant for the final discussion.

Restriction to Few Relevant Cases. The time frame of the work only permits a few cases to be analysed. Nevertheless, the aim is to find cases representative enough to draw interesting conclusions which might suggest further work in this area.

2.4.2. Theory vs Practice

The issues related to semantic interoperability are complex, and not less so in the disciplines of medicine and biomedical sciences, both because of the scale of the problem, with huge ontologies, and vast amounts of data and because of the high complexity and heterogeneity of the data. Although the problem under consideration to a large extent is theoretical by nature, it is beyond the scope and time frame of this work to gain new "high ground" at the theoretical level, i.e. to establish results of general validity.

Instead, concrete cases involving some archetypes will be analysed. The discussion will treat the possible limitations of the validity of the results based on these premises.

2.4.3. Other Issues

Interoperability is a necessary, but obviously not a sufficient condition to make the vision of the distributed EHR a reality. Other conditions involve security, usability, scalability, transaction management and other issues related to the system architecture. All that is not directly related to the problems of representation of medical information in such a way that semantic interoperability can be achieved, is beyond the scope of the present discussion.

2.5. Method

The nature of the relationship between the Data Structures in the EHR and a target terminology system is essentially theoretical, in the sense that it should be possible to gain an understanding of this relationship by theoretical means only. In this report, the theory is not established to that level. Instead, an experimental approach will be used, guided by theoretical considerations. Hopefully, the results of the experiments will serve as a motivator for further work in the theoretical direction.

\(^8\) http://www.ihtsdo.org/snomed-ct/
Theoretical Overview. The aim of this part is to give an overview of the background theory that will help to justify the hypothesis and the chosen approach.

Representative Samples. For the purpose of transformation and experimentation, archetypes will be chosen that are complex enough for doing meaningful reasoning with them, on the other hand simple enough for being able to do this without exhausting time and resource constraints.

Evaluating and Determining Tool set and Software Libraries. The area of interest has gathered considerable academic interest, and there are several open source tools and libraries available. The tool set to be used will be searched in the domain of open source software.

Transformation to OWL of the chosen EHR. For the OpenEHR Reference Model there already exist a "hand coded" OWL representation of a previous version of the RM Section 5.2, “Mapping the EHR to OWL”. This OWL-representation will serve as one of several starting points for an automatic procedure for the conversion of the RM to OWL, as discussed in Section 6.2, “Procedure Outline” and subsequent sections.

Validation of the Transformed Models. Having transformed the EHR models (Reference Model + chosen Archetypes) to OWL, we can apply a Reasoner on the combined models to validate their internal consistency.

Discussion, Evaluation, Generalisation. The results will be discussed and evaluated in view of the theory. Limitations of the method will be highlighted, and also possible generalisation.

2.6. Report Outline

To set the stage, a brief discussion on Medical Terminologies, Ontologies and their relationship to the EHR will be made, together with an introductory description of the Archetype based EHR and the SNOMED CT terminology. Relevant standards are covered, and the difficulties involved in linking the EHR to a terminology system are mentioned.

In the next chapter a brief description of architectural and theoretical topics are presented. Model Driven Architecture is discussed. Description Logics, the Semantic Web and Web Ontology Language are introduced.

There is considerable momentum in the Semantic Web research activities, and much endeavour is invested in trying to use OWL as a common denominator in the analysis of the problems at hand. The chapter on related work presents work closely related to the topics of this report, most notably the work done to express the OpenEHR RM and the archetypes in OWL.

Having prepared the ground, the steps involved in the practical work are described, as well as the outcome of this in relationship to the working hypothesis. Conclusions are drawn, and finally, future work is suggested.
3. Medical Knowledge and the EHR

Medical or medically related information can be partitioned into two categories: Information related to a specific Subject of Care, and that which is not. In the first case, it is in the realm of the Electronic Health Record, and in the latter case it falls into the category of medical knowledge. Roughly spoken, the EHR holds the facts about the diseases of an individual, while the other information is about what is (supposed to be) true for any individual about diseases, treatments medication regimens and so forth. The distinction is not so sharp, though. In the EHR as we know it today, there is much hidden or implicit knowledge. A decision on a particular treatment or a recorded diagnosis are often a function of implicit knowledge, that is, of knowledge that is assumed to exist but without any explicit and formal link to it.

When striving for semantic interoperability, it is a goal to separate facts from knowledge and then to make the relationship between them explicit by coupling entities in the EHR to entities in the knowledge base. The set of such couplings for an EHR system is called a Terminology Binding.

3.1. Recording Medical Knowledge

The types of information falling into the category of knowledge range over a wide spectrum, from loosely structured natural language documents (articles, textbooks etc.), to formal, machine processable artefacts of varying sophistication with the aims of classifying, organising knowledge, and at the highest level, even representing knowledge (cf. Section 3.1.2, “Ontologies”).

3.1.1. Terminologies

In its original meaning, terminology is the study of terms and their use. In the context of computers and Informatics, the term 'terminology' has come to denote human created resources (artefacts) that are collections of assignments of terms to corresponding definitions, and possibly also to other relevant information. So, a terminology is, at least in its more elaborated forms a sort of classification scheme. But even a classification can in itself be classified [PIDCOCK03]:

Types of Terminologies

- **Controlled Vocabularies.** Closed list of terms with unambiguous definition
- **Taxonomy.** A hierarchically organised controlled vocabulary (can be represented by tree structures)
- **Thesauri.** A taxonomy extended with other relationships than the hierarchical is_a relation (can be represented by directed acyclic graphs)
- **Ontology.** There are various definitions for an ontology, some of which will be given in the next section, and further explained in Section 4.2, “Description Logics”. In order to convey a rough meaning of the term, it can be defined as a thesaurus whose terms and expressions are constrained by a formal language, and where rules of logic govern the relationships in the model structure.

3.1.2. Ontologies

What is an ontology? The term originates in the field of philosophy, where it is set to denote a theory of the nature of existence. In computer science, the term denotes an artefact, (i.e. a human designed entity)
that *models some part of existence*. This definition is imprecise, but given in order to give a feeling for the link between its original meaning and the meaning such that it will be used in the rest of this work.

In [GRUBER93] we find the following definition of the term "ontology":

> A conceptualisation is an abstract, simplified view of the world that we wish to represent for some purpose. Every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualisation, explicitly or implicitly. An ontology is an explicit specification of a conceptualisation: A specification of a representational vocabulary for a shared domain of discourse — definitions of classes, relations, functions, and other objects.

The same author, about fourteen years later [GRUBER07] defines the term as follows:

> In the context of computer and information sciences, an ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members). The definitions of the representational primitives include information about their meaning and constraints on their logically consistent application.

The two definitions differ mostly in that the second one has omitted the reference to *conceptualisation*. This term is criticised by some scholars who want to see the term ontology to stick to the original notion linking it to *reality* (or existence) rather than to *concepts*. One may denote these different views on ontology as the *realist* vs. the *conceptualist* approach to ontologies. We are now ready for a third definition with even harder emphasis on the relation to *reality* [SPEAR06]:

> “...a representational artifact whose representational units are intended to designate universals in reality and the relations between them”

The basic idea is that ontologies are about what is general, structured and law-like in reality. Further, ontologies represent this generality not only by containing general or common terms representing universals, but also by capturing and explicitly representing the relationships that obtain amongst these universals.

**Ontologies and Logic.** A particularly interesting aspect of modern ontologies is their commitment to *description logics* which enables for the usage of *automatic reasoners* for the analysis of ontologies. A reasoner carries out such tasks as finding out if the ontology is inconsistent, which means that there is no possible "incarnation" of the model, and can also find "hidden" relationships in the ontology, i.e. relationships that that are not explicitly asserted by the modeller, but that must exist given those that are asserted. A reasoner is extremely useful when dealing with large ontologies. This will be further described in Section 4.2, “Description Logics”.

### 3.1.3. SNOMED CT

SNOMED CT is a Medical Terminology System. SNOMED CT is managed and Standardised by IHTSDO, which is an organisation supported by a number of member states, including Denmark and Sweden among the Nordic countries. The terminology contains more than 300 000 clinical concepts. Several national language translations exist, and a Swedish translation of the terminology is underway.

SNOMED CT is actually expressible in a subset of Description Logics, EL++ (cf. Section 4.2.4, “Classes of DL Languages” and Section 4.4.2, “OWL 2 Profiles”). The SNOMED official distribution comes with a script for converting the terminology to OWL, conformant to the OWL 2 EL profile. For this reason, it is also considered to be a medical ontology system, and is a candidate for experiments with terminology bindings in the follow-up to this work. However, there are issues with SNOMED CT from a Description Logics and Ontological point of view, which will be briefly discussed in Section 3.3.2, “The Interface between EHRs and Terminology Systems”.

---

3.2. The Electronic Health Record

The Electronic Health Record holds clinical data bound to a particular patient. The EHR can be defined as a persistent, longitudinal computer processable record of patient health information generated by one or more encounters in a care delivery setting. It includes all types of clinical patient bound information recorded over time, both manually created (e.g. in interaction with an EHR system) and generated by medical equipment (e.g radiology images, lab results).

The EHR has the potential to be supported by and support other care-related activities such as decision support and quality management, but this can be realized to full extent only when the semantic interoperability issues are solved. Interoperability issues of the EHR have long been an area of focus for special interest organisations, standards bodies and academic environments. I will point out some EHR research and standardisation activities of relevance to this report.

**EHR-fragment.** This frequently used term\(^\text{10}\) is informally described as a "well defined", "self contained" part of the EHR of a given subject of care. "Well defined" means that all necessary parts are present in order to make the set of entities constituting the EHR-fragment interpretable, and "self contained" means that it carries enough context information in order to be "linked in" in the appropriate part of the overall structure on the receiving side if transmitted from one EHR system to another. The concept is useful, because in the typical use case, it is not an entire EHR for a given patient that is communicated across systems, but rather a meaningful part of it, such as a medical discharge history, an anamnesis, a lab result etc.

3.2.1. The OpenEHR Foundation

The OpenEHR (Open Electronic Health Record) Foundation is a not-for-profit foundation with a long background history\(^\text{11}\). Its stated mission is to

- Making the interoperable, life-long electronic health record a reality
- Improving health care in the information society

It does this by:

- Developing open specifications, open-source software and knowledge resources
- Engaging in clinical implementation projects
- Participating in international standards development
- Supporting health informatics education

The foundation publishes and maintains a series of open specifications for the interoperability of EHRs. In recent years the result of their work has gained widespread interest and is finding its way into the standardisation committees.

The OpenEHR approach to the combined requirements of flexibility, rigour and interoperability is the so called *two level* modelling approach based upon the *Archetype*\(^\text{12}\) concept.

3.2.2. The Archetype Concept

An Archetype in an EHR context is a formal specification of a clinical concept represented as a data structure specification for an electronic health record fragment. The specification is expressed as a set of restrictions on a given RM. The model for thus constraining a Reference Model is defined by the *Archetype Object Model* (AOM).

---

\(^{10}\) The term *EHR-extract* is also often used to denote the same concept

\(^{11}\) [http://www.openehr.org/about/origins.html](http://www.openehr.org/about/origins.html)

The idea is to have an underlying generic but medical domain specific reference information model which is constrained to specific types of EHR-fragments through an Archetype definition.

So, an Archetype is an artefact that expresses a restriction on a "generic" and "medical domain specific" Reference (information) Model. Clearly, the terms "generic" and "medical domain specific" somewhat contradict each other. This should be interpreted pragmatically as "as generic as possible", and "as little domain specific as possible" in order to express our domain concepts without elaborated and clumsy constructions, and such that the RM remains stable over time, that is, the RM should not be susceptible to necessary changes as the domain knowledge increases.

There is (at least at the present state of understanding) no exact given optimal balance between these poles, but the extremes should indicate that there is a "golden middle way": Consider a RM which constitutes the set of letters and digits. Then each and every Archetype would be a grammar constraining which strings could be expressed for that Archetype. So we have a ultra-light and extremely generic RM and an Archetype model which is unmanageable by its complexity. On the other extreme, we could have a RM which tries to describe (give an information model for) as many of the clinical concepts as possible, and where the Archetypes are limited to "fine tuning" these by simple restrictions. This RM would be extremely susceptible to change when the domain knowledge evolves. The balance trade-off between genericity and domain specificness of the RM has to be found by heuristic. The fact that the OpenEHR RM and the RM of CEN/EN 13606 differ, while their Archetype model is similar is a clear indication of this.

Standardisation of archetype specifications constitute one fundamental prerequisite for EHR interoperability. The foundation maintains a growing Archetype library whose entries pass a formal review process before being accepted.

### 3.2.3. Standards Supporting the Archetype Concept

In 2002, CEN made a decision to revise its pre-standard ENV 13606 for health record communication. In that process the two level/Archetype approach was formally adopted, and is now incorporated in the final EN 13606 standard.

- **CEN.** EN 13606, "Health informatics - Electronic Health Record Communication" - a five part standard:
  - **Part 1, Reference Model.** Generic model for communicating part or all of an EHR between heterogeneous systems. This RM is basically a simplified subset of that of the OpenEHR RM.
  - **Part 2, Archetype Model Specification.** Specification of the OpenEHR originated method for creating clinical "information artefacts" by imposing constraints on the RM. These constraints can be expressed in a formal language, the ADL.
  - **Part 3, Reference Archetypes and Term Lists.** A set of model conversion archetypes, mapping to OpenEHR and to the HL7 v3 RIM Act classes. It also defines term lists specifying the set of values that particular attributes of the RM defined in part one may take.
  - **Part 4, Security Measures and Models.** For sharing access control, consent and audit-ability information of EHR communications.
  - **Part 5, Interface Specification.** Message and service interfaces to enable EHR and archetype communication.

- **ISO.** The ISO Technical Committee (TC) 215 is responsible for Health informatics standardisation. During the period 2008-2009 the first four parts of the EN 13606 have been formally adopted as ISO standards, and part five is registered for formal approval and is currently in the ballot phase, most likely to be approved upon in the near future, cf. [ISO35.240.80] "IT applications in health care technology"

---

13 [http://openehr.org/knowledge/](http://openehr.org/knowledge/)
The standardisation process, and status (as of 2008) is briefly described in [OPENEHR-CEN]. The difference with today’s status is mainly the progress made in the ISO standardisation process.

3.3. Linking the Health Record to Medical Knowledge

EHRs and Medical Terminologies have developed independently from each other, driven by different needs. The EHR evolved because of the immediate pragmatic need to record the medical history of the patient, while terminologies were developed in order to gather and systematise knowledge to be used in research, in decision support systems, for statistical analysis etc.

3.3.1. The Need for the Linkage

More recently, the need for semantic interoperability between different instances of EHR systems has become a driving force for terminology binding, the act of assigning semantical constructs from a medical terminology to entities in the EHR, thereby making the meaning of the recorded facts in the EHR explicit and formal, and in conformance to the actual "state of understanding”. This development will also serve the other purposes where "facts meet knowledge" mentioned above such as e.g. clinical decision support, because just as the EHR needs a knowledge based reference frame to make sense of the facts recorded in it, the knowledge oriented systems need access to the medical facts in order to evolve the knowledge.

3.3.2. The Interface between EHRs and Terminology Systems

When trying to describe the separation of concerns of the terminology system and the EHR, one often refers to the former as the "ontology" and to the schema of the latter as the "information model". This alludes to a clear cut and well defined distinction between the two, but that is not the case in todays systems.

In [USCHOLD03] the notion of a semantic continuum in the systems is established. On one extreme there is the situation where the semantics are totally implicit, and on the other, where the semantics are explicit and formal for machine processing. This tension is present in EHR/Terminology systems as well. There is considerable implicit semantics in the realm of the EHR, and even if all semantics in the Terminology system were formal and machine processable, a grey zone exists between the EHR and the terminology system where both make (implicit or explicit) semantic claims or assumptions regarding the same entities. This problem of redundancy and potential source of inconsistency is referred to as the boundary problem.

Work has been done in order to draw the limits of responsibility between SNOMED CT and the Archetype descriptions: In an NHS document from the "Connecting for Health” project (TermInfo project) [MARKWELL08A] of which [MARKWELL08B] is a compacted version, some principles and heuristic are presented in order to strike the good balance between what information is to be represented in the "information model" and "terminology system" respectively. But this is probably very hard to achieve unless the EHR is designed in view of a given terminology binding.
4. Theoretical Underpinnings

The fact that logic cannot satisfy us awakens an almost insatiable hunger for the irrational. (A. N. Wilson)

The need for a "semantic distillery" and for formal methods have been invoked in Section 2.2.4, “Problem Statement” and other places, leaving the definition of these terms to the intuition. Admittedly, calling for formalisms in itself calls for more precise definitions. Before going into the themes of Model Driven Engineering and of Description Logics, the concepts of models, and of formal methods will be introduced.

4.1. Formal Methods, Models and Modelling

4.1.1. Formal Methods

The following are two different definitions, both presented because of their different angle of view. The first definition is simpler, stricter and confined to computer science, while the latter emphasises that formal methods supplement human skills and experience, rather than replacing them:

Formal methods are mathematical techniques for developing computer-based software and hardware systems.\(^{15}\)

Formal Method: A group of analytical approaches having mathematically precise foundation which can serve as a framework or adjunct for human engineering and design skills and experience (U.S. National Institute of Standards and Technology)\(^{16}\)

The term is usually associated with different methods of various ambition levels, aiming at asserting or assessing program correctness, i.e. that an execution of the program yields correct output, for all valid input data, according to the specification. A requirement for this is that the specification itself is expressed in a formal way. Taken to its extreme, i.e. actually proving program correctness at all levels tends to be extremely complex and time consuming for programs and systems beyond the trivial. Maximal use of such methods are therefore seldom used unless there are extreme requirements on program correctness, for instance where there otherwise is a risk of massive loss life.

Our focus will instead be on the output itself of the programs: that the semantics of the data structures created and exchanged are consistent with some formal specification for them. For this purpose, Description Logics will be used. Given its foundations in mathematical logic, its usage satisfies the definitions above of a formal method.

4.1.2. Models and Modelling

EHR-systems (and software systems in general) are machine executable instantiations of artefacts representing some facet of the reality. The act of mapping those parts of the reality that are considered of interest onto a set of conceptual representations is called modelling and the result of the endeavour is one or more models.

When reasoning about compatibility and other issues between EHR-systems we actually analyse their models, and therefore, we want to be able to produce models of the system at various levels of abstraction. This leads naturally to a specific branch of Software Engineering, Model Driven Engineering (MDE), which views the entire development process as producing a series of models capturing various aspects of the system, and successively refining these models, down to the level where the models serve as close support to the implementation, or in some cases even down to the level where the model itself

\(^{15}\)http://formalmethods.wikia.com/wiki/Formal_methods

\(^{16}\)http://www.itl.nist.gov/div897/sqg/dads/HTML/formalmethod.html
4.2. Description Logics

4.2.1. History of DL

The history of Description Logics can be traced back to research in the field of Artificial Intelligence and Knowledge Representation, dating back to the seventies. The aim was to be able to develop systems with some “reasoning autonomy”, that is, systems being able to infer the implicit consequences from their explicit “knowledge base”, and then respond accordingly, hence the term ”intelligent” systems. Within this field there crystallised a research direction based on First Order Predicate Logic (FOPL).

But there was one serious obstacle, the reasoning algorithms were not tractable in many typical usage scenarios, i.e. they had a time complexity prohibiting their practical use. This led to intensive research for algorithms of reduced time complexity and other ways for overcoming or alleviating the problem. A breakthrough insight was that by reducing the expressiveness of FOPL in particular ways, one could identify tractable reasoning algorithms. The research was now directed at how to minimise the reasoning complexity, while retaining the maximum expressiveness. It turned out that dramatic reasoning improvements could be made while still retaining an expressiveness sufficient for many real world type problems. This in turn led to a boost in the research interest in this field, and the discipline of Description Logics (DL) was born. The following sections present a summary overview of DL and the applicability of reasoners on DL based models. For a comprehensive overview of the subject, covering historical roots, theory, reasoners, complexity of reasoning algorithms, reasoner optimisation as well as its usage in applications, [BAADER07] is a standard reference work. A cursory overview of the topic can be found in the presentation [HORROCKS06B], which also shows the role of OWL as a DL language.

4.2.2. Description Logics Concepts

Description Logics are a family of languages for Knowledge Representation. The need for different DL languages comes from the fact that different problems need different expressiveness, and as a rule of thumb, the less expressive the language, the better the reasoner performance. Thus, in order to maximise reasoner responsiveness, one would choose the smallest language with sufficient expressiveness for the task. The DL family of languages is a judiciously chosen set of subsets of first order predicate logic where the aim is to maximise expressiveness in such a way that their corresponding reasoner algorithms are tractable.

The fundamental building blocks of DL are concepts, roles and individuals. Furthermore, the DL languages provide a set of concept constructors and role constructors (collectively denoted constructors). A knowledge base contains a set of axioms that are statements that are supposed to be true in the domain. The semantics of these can be interpreted informally, using terms from set theory and logic, as follows:

**Concepts.** A concept (in OWL denoted class) is defined as a unary predicate, such that if it holds for a particular individual, then it is member of a set defined by the predicate. Thus, belonging to the concept (or being member of the class) is equivalent to set membership.

**Roles.** A role (in OWL denoted property) is a relation between two individuals and can be stated as a binary predicate such that if it holds true between two individuals, then they are related by the role.

**Individuals.** Individuals are named constants, they are the basic entities of the universe of discourse that we want to describe and organise.

**Constructors.** The constructors are the building blocks from which more complex concepts (and roles, but only concept constructors will be exemplified here) are built. New concepts are initially constructed from the atomic ones, i.e. concepts that are considered given, and which are not defined in terms of any other concept. From there on, new concepts can be built from other existing concepts, atomic or not. For the sake of illustration, the following table shows a few examples of concept con-
structors. Applying e.g. the concept constructor (5) yields the concept of those individuals that have only daughters (if any child at all).

<table>
<thead>
<tr>
<th>DL Syntax</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) $C_1 \cap \ldots \cap C_n$</td>
<td>Concept Intersection</td>
<td>$\text{Married} \cap \text{Child}$</td>
</tr>
<tr>
<td>(2) $C_1 \cup \ldots \cup C_n$</td>
<td>Concept Union</td>
<td>$\text{Man} \cup \text{Woman}$</td>
</tr>
<tr>
<td>(3) $\neg C$</td>
<td>Complement Of</td>
<td>$\neg \text{Man}$</td>
</tr>
<tr>
<td>(4) $\exists P.C$</td>
<td>Existential Quantification</td>
<td>$\exists \text{hasChild}\cdot \text{Human}$</td>
</tr>
<tr>
<td>(5) $\forall P.C$</td>
<td>Universal Quantification</td>
<td>$\forall \text{hasChild}\cdot \text{Female}$</td>
</tr>
<tr>
<td>(6) $\geq n P.C$</td>
<td>Min Cardinality</td>
<td>$\geq 3 \text{hasChild}\cdot \text{Male}$</td>
</tr>
<tr>
<td>(7) $\leq n P.C$</td>
<td>Max Cardinality</td>
<td>$\leq 1 \text{marriedWith}\cdot \text{Woman}$</td>
</tr>
</tbody>
</table>

**Axioms.** With the capability of describing concepts (and roles), the next step is to state facts about them, which is the same as stating relationships among them. Such statements are called axioms because we postulate their truth as a result of the modelling. Axioms fall into two categories: **Terminology definitions** and **World descriptions**. The former are "general" statements about the relationships among the concepts and roles in the domain and belong in the TBox ("T" for Terminology), and the latter are statements about the individuals in the domain, and belong in the ABox ("A" for Assertion). By analogy with a database, the TBox corresponds to the Schema, and the ABox to the data in the tables. The following illustrates this with some examples:

- **Terminology Definitions — The TBox.** General statements about concepts and roles, not involving individuals:

<table>
<thead>
<tr>
<th>DL Syntax</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) $C_1 \equiv \ldots \equiv C_n$</td>
<td>Concept Equivalence</td>
<td>$\text{Man} \equiv \text{Human} \cap \neg \text{Woman}$</td>
</tr>
<tr>
<td>(2) $C_1 \subset C_2$</td>
<td>Concept Specialization</td>
<td>$\text{Woman} \subset \text{Human}$</td>
</tr>
</tbody>
</table>

- **World Definitions — The ABox.** Statements about individuals, i.e. assertion of concept membership, and of relationship to other individuals:

  Woman(lisa), marriedWith(peter,lisa), Man(peter), hasChild(peter,jane)

These building blocks enables the construction of complex concept structures. Once built, the next step is to study the consistence of the constructed model.

### 4.2.3. Reasoning

As we have seen, a driving force behind DL is the availability of automatic reasoning services. A reasoner operates on the knowledge base by performing **logical inferencing**. The strict mathematical formalism enables the detection of implicit consequences from the explicit facts by application of logical rules. Some typical reasoning services on the knowledge base are outlined in the following:

**Satisfiability checking.** When there are no logical impediments for a concept in the knowledge base to contain individuals, then the concept is **satisfiable**. Conversely, when a concept definition contains a contradiction such that the concept cannot possibly have any individuals, then we say that the concept is **unsatisfiable**.

**Inconsistency checking.** A knowledge base is denoted as **inconsistent** when its axioms contain a contradiction which prevents it from having an interpretation where all axioms are simultaneously true. This is stronger than unsatisfiability, because a concept can be unsatisfiable without the entire knowledge base being inconsistent. An example of inconsistency is if one asserts that an individual belongs to a concept that is unsatisfiable.

**Classification.** A concept $B$ is said to **subsume** a concept $A$ if all individuals belonging to $A$ also belong to the concept $B$. Reasoners can compute the concepts that subsume a given concept as well as the classes that are subsumed by it. Therefore, the reasoner can infer entire subsumption hierarchies, and are therefore often called **classifiers**.

Reasoning services constitute a crucial part of the rationale for using DL when performing tasks related to semantic issues such as capturing and reasoning about the semantics of systems. When the
knowledge base is large, it is beyond the capacity of humans to infer all implicit consequences of the given axioms. [ALLEMANG08] provides a good overview over the workings and utility of reasoning services in the context of the semantic web.

### 4.2.4. Classes of DL Languages

The DL languages can be grouped by expressiveness, i.e. by the constructors available in the language. The following gives a summary overview of the principles for the classification and for the notation used.

The minimal language of interest is a language called \( \mathcal{A} \) (attributive language). The following construction rule defines the \( \mathcal{A} \) expressiveness. A and R represent the atomic concept and atomic role respectively, i.e. a concepts and roles not defined in terms of other concepts and roles:

\[
C, D \rightarrow \begin{cases} 
A & \text{(atomic concept)} \\
\top & \text{(universal concept)} \\
\bot & \text{(bottom concept)} \\
\neg A & \text{(atomic negation)} \\
C \cap D & \text{(intersection)} \\
\forall R.C & \text{(value restriction)} \\
\exists R.\top & \text{(limited existential quantification)} 
\end{cases}
\]

When the \( \mathcal{A} \) language is extended with negation of arbitrary concepts, it is called \( \mathcal{A}\mathcal{C} \). This language extended with transitive roles is abbreviated \( \mathcal{T} \), and forms an important base for many DL languages, described by their constructors, and prefixed with \( \mathcal{T} \):

- \( \mathcal{H} \) Role hierarchy.
- \( \mathcal{R} \) Limited complex role inclusion axioms. Reflexivity and irreflexivity Role disjointness.
- \( \mathcal{O} \) Nominals.
- \( \mathcal{T} \) Inverse roles.
- \( \mathcal{N} \) Cardinality restrictions.
- \( \mathcal{Q} \) Qualified cardinality restrictions.

As an example, The OWL 1.1 language has \( \mathcal{SHOIN} \) expressiveness, i.e. \( \mathcal{S} \) extended with hierarchical and inverse roles, nominals and cardinality restrictions. OWL 2 is an extension of OWL 1.1, and is denominated: \( \mathcal{SHOINQ} \) [HORROCKS06].

A DL language which has sufficient expressiveness for describing well known biomedical terminologies, such as SNOMED CT and polynomial time worst case reasoning complexity is \( \mathcal{ELC}++ \). For this reason there is a particular OWL profile (cf. Section 4.4.2, “OWL 2 Profiles”) associated with it [GRAU07].

### 4.3. The Semantic Web

The vision of the Semantic Web was briefly introduced in Section 2.2.3, “A General Problem”. The essential vision of the Semantic Web is the extension of the existing web in such a way that “semantic web aware” software applications may process it without human intervention. It is therefore called a web of data.

#### 4.3.1. RDF

RDF is an essential component in the building of the Semantic Web. The crucial idea of RDF is to reuse the identification mechanism of the web — the URI, as a mechanism to tag all “resources” (addressable entities) as well as their relationships. The target of one link (relationship) may be the source of another, and thus, the data spans directed graphs. The traditional web can also be modelled as a directed graph of interlinked documents, but unlike it, in RDF, both the source nodes as well as the target nodes and the links in the graph are labelled by their resource identifier, represented by the URI. The RDF-graphs
thus become computer processable meta-data that can be filtered, aggregated, etc. before accessing the data itself. If the meta data is appropriately structured, you get a "semantic" web.

4.3.2. RDF Schema

While RDF provides the backbone for structuring the data on the web as a labelled, directed graph, it lacks means for structuring the data at a higher level. RDF Schema (RDFS) provide the basic mechanisms for this by providing the notion of type (class) and property (relationship) hierarchies, and means for asserting that individuals belong to certain classes and are related by certain properties. This enables the construction of hierarchical structures with the possibility of some basic inferencing mechanisms.

4.4. OWL — Bringing DL to the Semantic Web

OWL Web Ontology Language is a knowledge representation language formally grounded in the theory of Description Logics.

The second characteristic of OWL is its compatibility with the Web and the set of web related technologies by adhering to the URI/IRI schemes for addressing and naming, and by providing exchange (serialisation) formats in XML, the de facto standard way of exchanging data on the Web.

The first version of OWL became a W3C recommendation in 2004. In the meantime, advances have been made, particularly in the field of optimising reasoner performance. This, together with some recurring demands from the user communities led to the definition of a slightly more expressive variant of the language, OWL 2 [OWL2DOC09]. OWL 2 Web Ontology Language became a formal W3C recommendation in late 2009. It is this version of the language which has been used in the present work.

4.4.1. OWL 2 Features and Building Blocks

The new features of OWL 2 are outlined in [OWL2NF09]. As seen in earlier sections, there are different classes of Description Logics languages, according to their expressiveness. OWL 2 belongs in the class $\mathcal{SHOIN}$ of Description Logics languages [HORROCKS06], a more expressive variant than its predecessor, which had $\mathcal{SHIQ}$ expressiveness (cf. Section 4.2.4, “Classes of DL Languages”). The extension was designed so as to provide useful additions to OWL-DL requested by users, while not sacrificing its decidability.

4.4.2. OWL 2 Profiles

OWL 2 provides the following predefined profiles, which are subsets of the language each with a particular trade-off in the balance between expressiveness and reasoner performance. The following is a summary of the profiles, the full definition is found in [OWL2PROF09].

**OWL 2 EL.** OWL 2 EL enables polynomial time [GRAU07] algorithms for the standard reasoning tasks. It has sufficient expressiveness for describing many large-scale ontologies such as SNOMED CT.

**OWL 2 QL.** OWL 2 QL is particularly suitable for applications where lightweight ontologies are used to describe and organise large numbers of individuals and where it is useful to access the data directly via relational queries, such as SQL.

**OWL 2 RL.** OWL 2 RL is suitable for applications where relatively lightweight ontologies are used to describe and organise large numbers of individuals and where it is useful to operate directly on data in the form of RDF triples.

4.5. Model Driven Architecture

The Model Driven Architecture (MDA) specification, managed by OMG, is a predominant initiative within the MDE discipline, and the Eclipse Modelling Framework (EMF) initiative is a set of MDA compliant open source tools and implementations which have largely contributed to the actual use of
the method. One core contribution of (and/or insight leading up to) the MDA was the M3 architecture, which take a systematic approach to describing a multi layered model of models.

### 4.5.1. Meta-models and The M3-Architecture

The concept of *meta model* is familiar to anybody working with software design, database design or any other modelling activity. For the program designer creating UML models, the UML language definition is the meta model. When creating a program, which can also be seen as a modelling activity, the programming language is the meta model. But the MDA defines an architecture based on a third level, the *Meta-meta model* (M3) level. Figure 4, “MDA M3-Architecture and Some Examples” illustrates this together with some examples. The bottom layer, M0 denotes the “system”, or the “reality”. It is *represented by* one particular model instance at M1 level. That model *conforms to* the meta model at level M2. The meta model describes the concepts that the model instantiates. It explains how to interpret the model. The model itself must only make use of concepts described in the meta model. Therefore, it is said to *conform to* the meta model. The next layer defines the meta model for the meta model, the *meta-meta-model*. It describes how to interpret the meta model. One could now be inclined to think that this would go on forever, but luckily not, it stops here. The meta-meta-model conforms to itself and thus breaks the further chaining. As far as I am aware, the reason for having exactly three levels is heuristics — at some level the meta model is describing so general terms that it can also describe itself, another layer would add no further meaning — and this level turns out to be the third in practice.

The examples in Figure 4, “MDA M3-Architecture and Some Examples” show the sufficiency of three levels for some typical cases: An XML-document (M1), which can be conceived of as a model representing some part of reality, conforms to a particular XML-schema (M2). This schema conforms to a singular instance of XML-schema, the meta-meta model which describes what an XML-schema can describe. Since it is also an XML-schema, it also describes itself. The other examples are analogous.

The architecture is often called a four level architecture, but in [BÉZIVIN03] (which also gives a good overview of the conceptual framework), the case is made for calling it a 3+1 architecture, because the nature of M0 is different from the other, it is not a model, but "Das Ding an Sich", which also leads to another type of relationship between M0 and the other layers, *represented by*, as opposed to *conforms to*.

![Figure 4. The MDA M3-Architecture With Modelling Examples From the XML, UML and Java Technical Spaces](image)

### 4.5.2. Model Transformations

MDA is about extracting models from an underlying system, and then study or mimic the system by studying and manipulating the created models. Therefore, in MDA, the concept of *model transformation* is central. An analogy is the production of maps, which is a form of extracting models from a "system" i.e. the reality. If the initial model captures e.g. topographic, geologic and economic information, model transformations can be applied accordingly to produce specialised maps for the different specialities with e.g. different colouring schemes. Figure 5, “A Generic Model Transformation” gives an overview of a Model Transformation as conceived in MDA.
The relation 'conforms to' is denoted \( c_2 \)

**Figure 5. A Generic Model Transformation**

The model transformation takes one model, \( M_a \) as input, and produces a new model \( M_b \) as output. The transformation itself is represented by a model \( M_t \). The meta-models for \( M_a, M_b \) and \( M_t \) are \( M_{Ma}, M_{Mb}, M_{Mt} \) respectively. The meta-models have one common meta-meta-model, \( MOF \). In order to express the transformation, the common meta-meta model is important, because it enables the description of the mapping between the known meta-models. In addition, identifying the mapping between the source and target meta-models \( M_{Ma} \) and \( M_{Mb} \) enables to define the transformation \( M_t \) as a model of the transformation from \( M_a \rightarrow M_b \).

The relevance of introducing MDA/MOF and MOF based Model transformations are the following:

- **Model transformation Conceptual Framework.** Analysing the model manipulations we aim at might require some technical framework outside of the ontology sphere. Since the work is about extraction of models from existing models, a model oriented engineering framework with "in-built" concepts such as model transformations seems suitable. It is therefore relevant to the ideas of the present work.

- **Ontology Definition Meta-model.** ODM is a specification that aims at marrying MDA engineering techniques with the Semantic Web. The specification states that it can be used to
  - interchange of knowledge among heterogeneous computer systems
  - representation of knowledge in ontologies and knowledge bases,
  - specification of expressions that are the input to or output from inference engines.

- **Related work and future work.** Some related work (Cf Section 5, “Related Work” use MDA techniques. Furthermore, the discussion will relate this work to MDA, and based on that, suggest some future investigation along this track (cf. Section 8.5, “The EHR from an MDA Perspective”).
5. Related Work

"When the time is ripe for certain things, these things appear in different places in the manner of violets coming to light in early spring." (Farkas Bolyai)

With the rapid proliferation of XML technologies, the great interest for Description Logics related methods within the biomedical sciences and finally, the increasing interest in Semantic Web technology, it is natural to find several initiatives pointing in the same direction. Some of these are of particular relevance to the present work and will be highlighted in the following.

5.1. Mapping XML to OWL

There exists a representation of the OpenEHR Reference Model in the form of a set of XML-schemata\textsuperscript{17}. A mapping from XML-Schema to OWL could thus serve as a mechanism for creating the target OWL representation of the RM.

The need for an XML to OWL mapping is of more general nature, since it would facilitate automatic conversion from XML documents to OWL, thus making the task of "ontologising" any existing XML-data less daunting (one has to keep in mind though that an artefact does not automatically become an "ontology" just because it is expressible in OWL). The abundance of XML data on the Web gives a great impetus to mapping from XML documents to OWL, when trying to enhance legacy data on the web to semantic representations.

There were several articles related to this theme available, of which one [ANICIC07] "Mapping XML Schema to OWL" was of particular interest, first because it focussed on XML-Schema itself, and not on XML data and secondly because it used a systematic mapping approach like the model transformations described in the previous section on MDA. The methods and mappings in this paper were not directly re-used for reasons described in Section 6.3, “Mapping of the OpenEHR RM to OWL”, but their availability contributed to the decision to transform the OpenEHR Reference Model to OWL by using the official XML-Schema based representations.

5.2. Mapping the EHR to OWL

Semantic methods and knowledge representation techniques have been used in the biomedical sciences before the Semantic Web was introduced. It is therefore no surprise that we find early adapters of Semantic Web technologies in in the field of biomedical sciences, and in medical informatics. The references regarding mapping the EHR and Archetypes to OWL have been prime motivators for the present work.

5.2.1. Mapping the OpenEHR RM to OWL

Pioneering work was done in 2004 [ROMAN04] when the OpenEHR Reference Model was converted to OWL by a manual procedure, using the official specifications of version 0.95\textsuperscript{18} and the Protégé tool. An update of this work was considered as an approach, but was dropped in favour of devising an automatic transformation (cf. Section 6.3, “Mapping of the OpenEHR RM to OWL”). This work nevertheless contributed valuable input to the project.

5.2.2. Mapping Archetypes to OWL

Once the Reference Model is transformed into OWL, a natural next step is to express the archetypes themselves in OWL. Since an archetype is a "constraint filter" on a concept in the RM, it will be expressed as the OWL representation of that concept, extended with OWL constraints specific to the particular archetype.

\textsuperscript{17} http://www.openehr.org/releases/1.0.2/its/XML-schema/index.html

\textsuperscript{18} http://www.openehr.org/releases/0.95/roadmap.html
The question is then how to express the archetype specific constraints, given the RM in OWL representation. This is described in "Mapping Archetypes to OWL" [KILIC06]. Those ideas are to a certain extent re-used in the present work as well (Cf Section 6.4.3, “The ADL to OWL Mapping”).

Several different activities aim at automatically transforming archetype definitions from ADL to OWL.

**The Archetype Ontologiser.** An application that uses the ehr2ont framework [EHR2ONT08] for translating OpenEHR archetypes into OWL, following the procedure outlined by [KILIC06]. It makes use of the OpenEHR ADL-parser (cf. Appendix A, Section 2, “OpenEHR Related Tools and Libraries”) for the generation of an Archetype Object Model, from which the OWL output is constructed. The process is described more in detail in [LEZCANO08]. That work also makes the case for the potential of combining the translated OpenEHR Archetype Definitions with a rule based approach using SWRL19 rules. They suggest that this could be used e.g. for triggering alerts in decision support or monitoring systems upon particular conditions in the data.

**LinkEHR** 20. In this project, an approach based on MDA based model transformation principles is applied. The aim is the OWL representation of clinical archetypes [FERNANDEZ09A].

In a related project, OWL has been used as a common bridging language for transformation between ISO 13606 and OpenEHR Archetypes [FERNANDEZ09B].

---

19 Semantic Web Rule Language: http://www.w3.org/Submission/SWRL/
20 http://linkehr.com
6. Practical and Experimental Work

A theory is something nobody believes, except the person who made it. An experiment is something everybody believes, except the person who made it. -Albert Einstein-

As seen in previous sections, there is a gravitation towards description logics based methods in the Terminology as well as in the Data and Information Model parts of the EHR. In order to get an appreciation of the usability of the "Ontology Oriented" approach (yet another OO method...) beyond its theoretical appeal, the discussion will be complemented with practical work and experiments.

The aim is to transform the EHR models to an OWL based ontology representation in a sequence of steps. Assuming then that the Terminology System also has been or can be transformed into an OWL based representation, the groundwork for aligning these ontologies has been laid. This will enable the application of a reasoner with the aim of checking the consistency of all the models combined. These issues will be elaborated in Section 7, “Discussion”.

6.1. Open Source Tool Set

There is a need for software tools covering functionality such as XML editing and transformation, Ontology manipulation as well as reasoning and consistency checking. All tools used in relation to this work is open source. There are several reasons for this:

• **The Semantic Web is Mainly Driven by Open Source.** The first Semantic Web tools originated in the academia, and most of these have been under continual development and continue representing the "state of the art" of this technology

• **Need for Usage of OWL 2 Tools.** For reasons that will be elaborated on in the following, the new features in OWL 2 were deemed to be of such value that the choice was made for using this brand new version of the OWL language. OWL 2 [OWL2DOC09] became a formal W3C recommendation as recently as October 27th 2009, and only open source tools granting access to their development repositories support the latest version

• **Flexibility and Convenience in Development.** Having access to a broad set of high quality open source tools, many of which with very specialised functionality, makes it possible to combine them to get a tool chain configuration optimised for the task at hand

6.2. Procedure Outline

The following is a brief outline of the steps involved. Each step is further described in subsequent sections.

![Figure 6. Tool Pipeline](image)
• **Creation and Validation of RM Sub ontology.** The reference model (RM) is converted to OWL. The validity and consistency of this sub ontology is verified in a series of steps. First the syntax is checked with XML validation tools. Then the ontologies is loaded into a test and conversion program, checking that the input is valid OWL 2, and also determines its OWL profile (cf. Section 4.4, “OWL — Bringing DL to the Semantic Web”). In the last step, the created sub ontology is checked for consistency by means of a reasoner.

• **Conversion of a Chosen Set of Archetypes to OWL.** This step involves finding some representative archetypes, and convert them to OWL 2 according to a defined mapping.

• **Consistency checking and Validation of the Combined Ontology.** Since both the RM and the chosen Archetypes now have their representation in OWL 2, this step amounts to running the reasoner on the combined model. This enables us to compare the validity checking of archetypes as performed by a compliant OpenEHR system with the corresponding checks by an OWL reasoner upon an OWL 2 based representation of the same models.

### 6.3. Mapping of the OpenEHR RM to OWL

#### 6.3.1. Initial Approach

The initial idea was to reuse the pioneering work done by Isabel Román [ROMAN04], whereby the RM was mapped into OWL by a manual procedure, using the official OpenEHR RM specifications and the Protégé tool (cf. Appendix A, Section 3, “Semantic Web Related Tools and Libraries”. This work was based upon a previous version of the RM (0.9.5), as well as on a previous version of OWL (1.0). However, a choice was made early on to choose OWL 2 as the target language for the transformation, and not OWL 1.x. The reasons for this are presented below. Given the significant changes on both the source side (RM 0.9.5 to RM 1.0.2) and target side (OWL 1 to OWL 2) of the transformation, a manual procedure for updating the models was deemed to be too time consuming and error prone, in addition to its lack of re-usability. The initial effort in this direction was nevertheless of great utility because of the insights given by the study of the first RM to OWL mapping, and also for gaining familiarity with the Protégé-OWL tool.

#### 6.3.2. Automatic Transformation — Preparatory Considerations

An approach based on transformation from XML-Schema to OWL was chosen, since a series of XML-Schemata representing the RM are available on the OpenEHR web site [22], and because related work indicate the feasibility of such an approach (cf. Section 5.1, “Mapping XML to OWL”).

As a prerequisite for starting the programming, some preparatory work was done. This consisted mainly of deciding which version of OWL to use — the brand new OWL 2, or the more mature OWL 1.1. Furthermore, there exist several serialisation syntaxes for OWL, and one of them had to be chosen as the target for the transformation. With these questions answered, we could proceed to choose the appropriate tool set for performing transformations and validating the result.

• **Target Language: OWL 2**

OWL 2 Web Ontology Language (OWL 2), which was released as a W3C Recommendation as late as October 27, 2009, was chosen as the target language in favour of the older version, OWL 1.1. There were some risks involved with this choice, since the most important tool adapted to OWL 2, the OWL API v.3.0 [HORRIDGE09] (cf. Appendix A, Section 3, “Semantic Web Related Tools and Libraries”) was still a “moving target” with respect to full OWL 2 compliance, and snapshots from the development repository had to be used. Nevertheless, this risk was taken, since all necessary and most of the important new features of OWL 2 were already implemented in the OWL API, and the new features of OWL [OWL2NF09] were considered of great importance, if not even necessary, in order to obtain a satisfactory mapping mechanism. Among many of the new features, the following are particularly important for us:

---

21 http://www.w3.org/TR/xmlschema11-1/, http://www.w3.org/TR/xmlschema11-2/
22 http://www.openehr.org/releases/1.0.2/its/XML-schema/index.html
• **Property Qualified Cardinality Restrictions**

  OWL 1 does not provide this feature, whereas OWL 2 does. This means that when restricting the cardinality of a property, one may also restrict the property such that its range is of a specified type.

• **Extra Data types and Data type Facets**

  OWL 2 provides new capabilities for data types, supporting a richer set of data types and restrictions of data types by facets, as in XML Schema

• **Property chain inclusion**

  OWL 2 allows defining a property as the composition of several properties which is useful when modelling e.g partOf relations.

• **OWL/XML syntax**

  With OWL 2 there is also defined a serialisation syntax based on XML Schema. Hence, to take advantage of this syntax (see subsequent paragraph), OWL 2 is a mandatory choice.

• **Target Syntax: XML_Serialisation OWL/XML**

  Prior to OWL 2, the only XML based OWL syntax available was RDF/XML. This format is relatively "low level" and allows for a series of syntactical shortcuts, making the task of transformation to and from it rather hard. In contrast, the newer OWL/XML syntax that is available for OWL 2 is specifically designed for improving the interoperability with XML based tools and languages, for example XSLT/XPath (cf. [OWL2XML09]).

  The "XML-tool-chain friendly" property of OWL/XML made it a natural primary candidate as the target OWL syntax. OWL/XML is supported by the OWL API v3.0. It was therefore decided to use OWL/XML as target syntax for the XML-Schema to OWL 2 transformation. As an additional benefit, the OWL/XML syntax closely mirrors the functional style syntax of OWL 2 (cf. [OWL2FSS09]) and is significantly more "human friendly" in reading than RDF/XML.

  The generated OWL/XML will be read by an application which uses the OWL API, (cf. Appendix A, Section 1, “XML Transformation Tools and Libraries”) and at this stage, an ontology loaded from an OWL/XML source can be output in any of the supported syntaxes of the OWL API, including RDF/XML, which is convenient for interoperability with other OWL based tools, since many of them do not yet support OWL/XML.

  In Section 5.1, “Mapping XML to OWL” there are some examples of previous work related to mapping from XML-Schema (or XML instance documents) to OWL. These have inspired the work, but the mapping presented here is done from scratch. The reasons for this are mainly that these other mappings use OWL 1 as target language, and map to OWL/RDF, neither of which we do, as explained above.

### 6.3.3. Transformation and Validation Tools

Since both the source syntax (XML-Schema) and the target syntax (OWL/XML) are XML documents, we use XSLT Transformations\(^{23}\) which is a language designed for this purpose. The tool set for XSLT processing and validation is described in Appendix A, Section 1, “XML Transformation Tools and Libraries”.

An XSLT program named *xsd2owl* was created for the transformation of the RM from XML-Schema to OWL/XML. For validation of the output from this program, a small Java test program, *Owlconv*, making use of the OWL API v.3.0 was developed (cf. Appendix A, Section 3, “Semantic Web Related Tools and Libraries” and Appendix B, *Lab Artefacts*).

\(^{23}\) [http://www.w3.org/TR/xslt](http://www.w3.org/TR/xslt)
6.3.4. Transformation Outline

For a full account of the details of the mapping, I refer to the source file xsd2owl with embedded comments\textsuperscript{24}. The OpenEHR RM specification is originally defined as a set of annotated UML class diagrams. The main principle of the mapping is to map the original UML (in XML-Schema representation) classes to OWL classes, and the attributes to OWL properties. The semantics of these concepts are different in the technical space of Description Logics as compared to the technical space of Object Orientation, so care has to be taken to conserve the overall intended semantics.

**Declarations.** OWL 2 requires that all entities appearing in an ontology are declared, which essentially implies naming them, and asserting their type. Thus, we locate all elements from the XML-schema that will be mapped into OWL 2 entities and declare them. This procedure is detailed below.

**Class and Restricted Data Type Definitions.** There are two types of class definitions in the XML-schema source: Restricted Data type definitions, and “Ordinary” class definitions. The former are refinements of standard data types, the restrictions consist of defining value ranges or patterns which additionally constrain these standard types, and these constrained primitive types are defined as an owl data type. “Ordinary” Class definitions are mapped to OWL classes with the same name.

**Property Definitions.** In the UML specification and in the corresponding XML-Schema mapped from it, object oriented semantics applies: The classes contain attributes “belonging” to their defining class. When a class is instantiated, (i.e. an object is created), the attributes may or may not be instantiated as well, depending on the specification. At run time, attributes may refer to primitive types (integer, real etc) or they may refer to other objects. The type of values or objects that the attributes may reference is defined statically (i.e. at design time). The attributes are statically typed.

OWL, on the other hand, does not have the concept of attributes belonging to instances of classes. Instead it defines properties as relationships between pairs of individuals (OWL term for "instances" or "objects"). It is possible to specify domain and range restrictions on a property. A domain restriction constrains the type of individuals that can exist on the source side of the relationship to the specified type. Correspondingly, a range restriction constrains the type of individual that can exist on the target side to the specified type. If we conceive of a property as a named and directed arc representing a relationship between two individuals, the domain restriction is a restriction on the outgoing side of the arc, and the range restriction is on the incoming side. These restrictions make it possible to map the attributes in the OO semantics to the properties of OWL: each attribute is mapped to an OWL property such that the “owning class” of the attribute is mapped to a domain restriction on the corresponding property, and the type of the attribute is mapped to a range restriction on it. In the OO world, an attribute is normally in the lexical scope of its enclosing class. Therefore, there may be many attributes with the same local name, but with different owning class, without causing ambiguity. This is not the case for OWL, where properties are defined independently of classes. In order to resolve this source of ambiguity, all properties are named by taking the name of the attribute from which they are mapped, and prefixing them with the name of the "owning class” of the attribute.

6.3.5. Mapping Details

As an example, we will show the mapping of the OpenEHR class \texttt{DV\_TEXT} from XML-Schema to OWL 2. The class inherits from \texttt{DATA\_VALUE}, and contains some attributes both of primitive and object valued types, with different cardinality restrictions. Example 1, “\texttt{DV\_TEXT} in XML-Schema” and Example 2, “\texttt{DV\_TEXT} in OWL” show how the elements are mapped. The list of annotations between the examples (Source and Target Elements of the \texttt{DV\_TEXT} Mapping) link the source and target elements of the mapping.

\textsuperscript{24}http://inherit.se/semweb_ehr/xsd2owl.xsl
Example 1. Extract of the OpenEHR Class DV_TEXT in XML-Schema Representation

Annotations Linking the Source and Target Elements of the DV_TEXT Mapping

1. The declarations of class DV_TEXT, subclass of DATA_VALUE
2. The value attribute with no cardinality restrictions in the source, mapped to exactly 1 in the target.
3. The hyperlink attribute, with a ‘minOccurs=0’ restriction, mapped to a [0..1] restriction in the target.
4. The mappings attribute, with a ‘minOccurs=0’ and a ‘maxOccurs=unbounded’ mapped into a universally quantified restriction, expressing that ‘all values of this property (if any at all) should be of type TERM_MAPPING.

Example 2. DV_TEXT Extract Mapped into OWL 2 OWL/XML Serialisation Syntax
The resulting OWL2 entities in the target mapping are Classes, Data types, Data Properties and Object Properties. (OWL also has Annotation Entities and Individuals but these entities do not figure in the mapping here).

The attributes of DV_TEXT are mapped into OWL properties of the same name, but prefixed with the name of the enclosing class and a hyphen: ‘DV_TEXT-‘. The attributes of basic type, such as xs:string are mapped into Data Properties, and the attributes of non primitive types, i.e. object valued attributes, are mapped into Object Properties.

The cardinality constraints are expressed differently in XML-Schema and OWL. In XML-Schema the default cardinality is one, i.e. omitting a cardinality constraint means a cardinality of one, while in OWL it means any cardinality (0..*) since nothing is said about it.

A class can conceived of as the set of individuals that belongs to it, and one way of defining a class in OWL is by asserting its name, and then to assert it to be equivalent to an anonymous class defined in terms of a set of restrictions. The OWL definition of the class DV_TEXT, shown in Example 2, “DV_TEXT in OWL” below can be stated in natural language as:

"The class DV_TEXT is the set of individuals such that they:
• belong to the class DATA_VALUE, and
• have exactly one property with domain DV_TEXT and range xs:string and
• have at most one property with domain DV_TEXT and range DV_URI and
• etc..."

6.4. Conversion of Selected Archetypes to OWL

At this stage, the OpenEHR Reference Model (RM) has a representation in OWL 2. The next step is to choose some OpenEHR Archetypes for transformation into OWL 2. When both the RM and the Archetypes coexist in the same "technical space" of OWL and Description Logics, we are in a position to use an OWL 2 reasoner for checking the validity of the Archetypes against the RM.

6.4.1. Manual Transformation

As seen in Section 5.2, “Mapping the EHR to OWL”, the transformation of OpenEHR Archetypes to OWL has already been done in an automatic process [LEZCANO08] by means of an application called the "Archetype Ontologiser" in the ehr2ont framework25. This is done by applying a transformation on a runtime instantiation of the Archetype Object Model, generated by the OpenEHR ADL parser (cf. Appendix A, Section 2, “OpenEHR Related Tools and Libraries”.

The reasons for not reusing the result of that work in the present work was mainly that it builds on an older mapping of the RM to OWL [ROMAN04]. This mapping is based in turn on older versions both of the RM (0.95), and of OWL (OWL 1.1). Since there are considerable changes in the current version of the OpenEHR RM (1.0.2), and since OWL 1.1 lacks some important features that are highly useful for some aspects of the mapping, it was not an option to use these versions of the RM and of OWL as a basis for this work. On the other hand, it was deemed to be beyond the time and resource frame to bring the ehr2ont implementation in sync with the requirements of the present work. Nevertheless, the existence of an earlier automatic transformation procedure is a strong hint on the feasibility of the method, and backs up the hypothesis of the present work. But since to our knowledge there was no automatic transformation available meeting our requirements, a task of manually transforming the archetypes to OWL 2 was undertaken.

6.4.2. Choice of Archetypes

Because of the tediousness of a manual transformation procedure, it is important to choose archetypes that strike a balance between being on one hand simple enough for the task to
be completed with reasonable effort, and on the other hand complex enough for the purpose of "proof of concept", i.e. for demonstrating subtleties in the validation process. For our purposes, the archetypes openEHR-EHR-OBSERVATION.BODY_WEIGHT.v1 and openEHR-EHR-OBSERVATION.BODY_WEIGHT_BIRTH.v1 were chosen. They are relatively simple, and the latter is a further constraint of the former. The first will thus be checked for consistency against the Reference Model, and the latter will be checked for consistency against the former. Furthermore, there is a design problem which make them incompatible with each other, which will serve as a test for the validation procedure.

6.4.3. The ADL to OWL Mapping

The mapping of archetypes to OWL follow the guidelines of [KILIC06], with some modifications. Firstly, the Kilic & al. mapping does not use the class prefixing method when mapping from attributes to properties. This may work for small ontologies, but with larger ontologies, the probability of name clashes increase. Furthermore OWL 2 has new features allowing for the capture of more of the semantics from the ADL, especially refined data type definitions from XML-Schema, and qualified cardinality restrictions which make it possible to capture the occurrences restrictions from ADL. The main principles of the mapping build upon those of mapping the Reference Model to OWL, and are as follows.

- The root class is created as a subclass of the concept it restricts. Thus, an archetype is represented by an OWL class, being a either a subclass of the RM concept that the archetype constrains, or a subclass or another archetype that is to be further constrained. The class BODY_WEIGHT, for instance, is created as a subclass of the concept OBSERVATION from the Reference Model.

- Cardinality and Occurrences constraints in ADL are combined to qualified cardinality constraints in OWL, such that the cardinality on the target side is the most restricted cardinality of the combined attribute and occurrences constraints.

- Each restricted ADL attribute at each level leads to the creation of a subclass of the class representing the range of the restricted attribute on the target side. This is repeated recursively down the hierarchy.

Thus, for instance, the restriction on the 'data' attribute of the class OBSERVATION leads to a new subclass, BW_HISTORY, and the corresponding property 'OBSERVATION-data' on the target side, is range restricted to BW_HISTORY.

The Example 3, “BODY_WEIGHT in ADL” and Example 4, “BODY_WEIGHT in OWL 2” show the mapping of the BODY_WEIGHT archetype from ADL to OWL 2. The "human friendly" Manchester Syntax is used for clarity, and for comparison with OWL/XML syntax, shown in Example 2, “DV_TEXT in OWL”. The list of annotations between the two examples (Source and Target Elements of the ADL2OWL Mapping of BW) link the source and target elements of the mapping.

```
archetype (adl_version=1.4)
openEHR-EHR-OBSERVATION.body_weight.v1

concept
[at0000] -- Body weight
definition
OBSERVATION[at0000] matches { -- Body weight 1
  data matches {
    HISTORY[at0002] matches { -- history 2
      events cardinality matches (1..*; unordered) matches {
        EVENT[at0003] occurrences matches (0..*) matches { -- Any event 3
          data matches {
            ITEM_TREE[at0001] matches { -- Simple 4
              items cardinality matches (1; unordered) matches {
                ELEMENT[at0004] matches { -- Weight

Example 3. Extract of Archetype BODY_WEIGHT in ADL
```
Linking the Source and Target Elements of the ADL to OWL Mapping of BODY_WEIGHT

1. OBSERVATION.data restriction
2. OBSERVATION.HISTORY.events restriction
3. OBSERVATION.HISTORY.EVENT.data restriction
4. OBSERVATION.HISTORY.EVENT.ITEM_TREE.items restriction

Class: <bw:BODY_WEIGHT> 1
   EquivalentTo:
   <rm:OBSERVATION>
   and (<rm:OBSERVATION-data> exactly 1 <bw:BW_HISTORY>)
   and ...

Class: <bw:BW_HISTORY> 2
   EquivalentTo:
   <rm:HISTORY>
   and (<rm:HISTORY-events> some <bw:BW_HISTORY_EVENT>)
   and (<rm:HISTORY-events> only <bw:BW_HISTORY_EVENT>)

Class: <bw:BW_HISTORY_EVENT> 3
   EquivalentTo:
   <rm:EVENT>
   and (<rm:EVENT-data> exactly 1 <bw:EVT_DATA_ITEM_TREE>)
   and ...

Class: <bw:EVT_DATA_ITEM_TREE> 4
   EquivalentTo:
   <rm:ITEM_TREE>
   and (<rm:ITEM_TREE-items> exactly 1 <bw:DATA_EL_AT_0004>)
   and ...

Example 4. Extract of Archetype BODY_WEIGHT in OWL 2 Manchester Syntax"

6.5. Some issues

When performing the Archetype to OWL mapping of the archetypes BODY_WEIGHT and BODY_WEIGHT_BIRTH, some issues regarding validity constraints and archetype specialisation were identified. Some are trivial errors, and some more problematic, indicating an underlying need for clarification or correction in the specification.

6.5.1. Archetype Specialisation

The principles of archetype specialisation are that an archetype either constrains

• a class from the underlying reference model

• another archetype

In both cases, the constraints are either on

• attributes as cardinality or existence constraints

• object instances as occurrences constraints

• object instances as type constraints, implicitly implying that the restricted type must conform to the type to be restricted, i.e. being in a subclass relationship to the latter
We have not located the official exact semantics of archetype specialisation, but the following is an excerpt from the OpenEHR wiki:

**Specialisation Semantics**

1. A non-specialised (i.e. top-level) archetype defines an instance space that is a subset of the space defined by the class in the reference information model on which the archetype is based.

2. A specialised archetype can specialise only one parent archetype, i.e. single inheritance.

3. A specialised archetype defines an instance space defining the following elements: unchanged object and attribute constraints inherited from the parent archetype; and one or more:
   - redefined object constraints, that are proper subsets of the corresponding parent object constraints;
   - redefined attribute constraints, that are proper subsets of the corresponding parent attribute constraints;
   - extensions, i.e. object constraints added to a container attribute with respect to the corresponding attribute in the parent archetype, but only as allowed by the underlying reference model.

4. All elements defined in a parent archetype are either inherited unchanged or re-defined in a specialised child.

5. Specialised archetypes are expressed differentially with respect to the parent, i.e. they do not mention purely inherited elements, only redefinitions and extensions.

6. Redefinition cannot remove an object constraint, only narrow it to a reduced instance space.

7. Extensions always define an additional subset of the instance space defined by the reference model element being extended (i.e. to which the 'new' objects belong). The extension capability allows archetypes to remain extensible without having to know in advance how or if they will be extended.

A closer look at the BODY_WEIGHT (BW) and BODY_WEIGHT_BIRTH (BWB) archetypes reveals some violations of the specialisation rules, referring to the specifications above.

**BW/BWB constraint violations on complex types.** In the BW archetype the type ITEM_STRUCTURE is constrained to ITEM_TREE in a couple of places. One of these is further constrained to ITEM_SINGLE, and another to ITEM_LIST in BWB. Now, both ITEM_SINGLE and ITEM_LIST are siblings to ITEM_TREE and not in a parent/child relationship. Interpreting “reduced instance space” as meaning a subclass relationship, this is a violation of pt. (6) above.

Also, it is unclear what is the point in these restrictions. Representing data in a tree structure rather than in a list structure does not necessarily change the semantics, both can be ordered or unordered collections, with or without duplicates. Furthermore restricting ITEM_TREE to ITEM_SINGLE seems like yet another facility for constraining the cardinality. Now, the ADL already has two such mechanisms (cardinality and occurrences), and introducing a third one by superimposing the same type of semantics in the definition of the collection classes themselves, is redundant (and confusing).

The workaround used in the mapping was simply to ignore these redefinitions.

**BW/BWB constraint violation on simple types.** In the BW archetype, the ELEMENT[at0004] allows for two alternative representations of DV_QUANTITY, one with the units in 'kg' and the other

---

26 http://www.openehr.org/wiki/display/spec/openEHR+Templates+and+Specialised+Archetypes
with 'lb'. Now this is restricted in BWB, such that the `DV_QUANTITY` with units 'kg' is further range restricted, as expected, but in the other alternative the units are changed to 'gm'. Since 'gm' is not in "a subset of the instance space" of 'lb', this is an error. We will return to this when discussing the validity checking in the next section.

6.5.2. Cardinality and Occurrences Constraints

The following is an excerpt from section 5.3.4.2 of the ADL 1.4 specification regarding Occurrences constraints:

A constraint on occurrences is used only with cADL object nodes (not attribute nodes), to indicate how many times in runtime data an instance of a given class conforming to a particular constraint can occur. It only has significance for objects which are children of a container attribute since by definition, the occurrences of an object which is the value of a single-valued attribute can only be 0..1 or 1..1, and this is already defined by the attribute existence.

...Where cardinality constraints are used (remembering that occurrences is always there by default, if not explicitly specified), cardinality and occurrences must always be compatible. The validity rule is: VCOC: cardinality/occurrences validity: the interval represented by: <the sum of all occurrences minimum values> .. <the sum of all occurrences maximum values> must be inside the interval of the cardinality.

**BW VCOC validity rule violation.** Consider the following extract from the BW archetype:

```plaintext
HISTORY[at0002] matches { //history
events cardinality matches {1..*; unordered} matches {
  EVENT[at0003] occurrences matches {0..*} matches { //Any event
data matches {
    ITEM_TREE[at0001] matches { //Simple
}
}
}
```

This violates the VCOC validity rule, since [0..*] is not a subsegment of [1..*]. In the mapping, this was just corrected to the most restrictive cardinality. When devising an automatic procedure for the transformation from ADL to OWL 2, one has to assume that this type of error is caught in an earlier step by the OpenEHR validation tools.

6.6. Consistency and Satisfiability Checking

The last steps in the pipeline are to check the transformed models for consistency. To do this, we make use of a reasoner to classify the models (cf. Section 4.4, “OWL — Bringing DL to the Semantic Web”). There are normally various alternative open source reasoners available, (cf. Appendix A, Section 3, “Semantic Web Related Tools and Libraries” each having different performance characteristics, depending on the profile of the models to classify. The feature set of the reasoners may also vary, but the basic functionality is the same, to determine if a given model is consistent and if so, identify classes that are unsatisfiable, if any.

We used the HermiT reasoner, developed at the Computer Laboratory of the University of Oxford. HermiT was the first (and at the time of the writing, the only) reasoner which was ready for the new OWL API v3.0.

A Java program, Owlconv, was developed as a client to the OWL API for the purposes of compliance reporting, validation and for conversion from one OWL 2 serialisation syntax to another. The script Owlconv.sh is a wrapper script around this program, and it can be run with or without reasoning services activated. Owlconv was configured to use HermiT as reasoner.

---

27 [http://www.openehr.org/releases/1.0.2/architecture/am/adl.pdf](http://www.openehr.org/releases/1.0.2/architecture/am/adl.pdf)
29 [http://web.comlab.ox.ac.uk/](http://web.comlab.ox.ac.uk/)
6.6.1. Validating the Reference Model

The RM had previously validated as syntactically valid OWL 2, using the XML-Schema validation tools. Furthermore, when running Owlconv (without reasoning service) on it, the RM was classified as a valid OWL DL ontology.

However, when running Owlconv with reasoning of HermiT activated, we got a run time error. Searching for the cause of the problem, it turned out that there was one particular FacetRestriction that HermiT did not swallow. The source of this restriction was the following (from the XML-Schema RM source):

```xml
<xs:simpleType name="matchString">
  <xs:restriction base="xs:string">
    <xs:pattern value="\?"/>
    <xs:pattern value="&lt;"/>
    <xs:pattern value="&gt;"/>
  </xs:restriction>
</xs:simpleType>
```

The source of the problem was the patterns '&lt;' and '&gt;' that HermiT did not accept. When removing one particular restriction, the only one containing these pattern strings, the Reference Model validated without problem. However, the validation now took almost two hours (on an AMD dual core 64 bits machine with 2Gb of memory) to complete. When removing all FacetRestrictions on the data properties from the transformation, the validation time was reduced to roughly 90 seconds (We will return to the issue of performance in Section 7, “Discussion”). For the demonstration purposes of the present report, these facet restrictions were not deemed to be essential. Removing them implied weakening some axioms (which means that the semantics also is weakened) regarding restricted primitive data types. These would not alter the points made in our experiments. Therefore, in the following example runs, the FacetRestrictions are removed in order to get manageable elapsed times for the reasoner.

6.6.2. Validating Archetype BODY_WEIGHT Against the RM

When running Owlconv, we get the result shown in Example 5, “Validating BODY_WEIGHT”:

```
rol@hamar:~$ ../Owlconv.sh 'bw#' '-v'
Adding entry to IRIMapper: http://some-iri/iriprefix/bw#...  
Adding entry to IRIMapper: http://some-iri/iriprefix/rm#...  
Loading ontology: http://some-iri/iriprefix/bw#  
Ontology with format: OWL/XML loaded
Profile violation report, http://some-iri/iriprefix/bw#: OWL2 Profile 
  No violations found in profile.
  No violations found in profile.

HermiT reasoner preparing...  
Building the class hierarchy...  
  ... finished  
Classifying object properties...  
  ... finished  
Classifying data properties...  
  ... finished
Consistent: true  
There are no unsatisfiable classes
```

Example 5. Validating archetype BODY_WEIGHT against RM

The output report shows the result of running validation on the combined RM and BODY_WEIGHT ontologies. The first part of the report are results of checks performed by services in the OWL API itself.
The loaded ontology is validated as an OWL ontology, and it also satisfies the additional restrictions making it compliant to the OWL DL profile. This has implications on the tractability of the validation process (cf. Section 4.4, “OWL — Bringing DL to the Semantic Web”). The last part of the report shows the result of applying the reasoner. The combined ontology is consistent, and furthermore, all classes are satisfiable, i.e. from a logical point of view there exists no class in the ontology such that it cannot contain any individuals. The BODY_WEIGHT archetype, represented in the ontology by an OWL class with the same name, is thus satisfiable, meaning that it is consistent with the RM.

6.6.3. Validating Archetype BODY_WEIGHT_BIRTH against BODY_WEIGHT and RM

In a next step, Owlconv is applied to the ontology obtained by combining the Reference Model with the archetypes BODY_WEIGHT and BODY_WEIGHT_BIRTH. We get the result shown in Example 6, “Validating BODY_WEIGHT_BIRTH”:

```
rol@hamar:~/$ ../Owlconv.sh 'bwb#' '-v'
Adding entry to IRIMapper: http://some-iri/iriprefix/bw#/...
Adding entry to IRIMapper: http://some-iri/iriprefix/rm#/...
Adding entry to IRIMapper: http://some-iri/iriprefix/bwb#/...
Loading ontology: http://some-iri/iriprefix/bwb#
  Ontology with format: OWL/XML loaded
Profile violation report, http://some-iri/iriprefix/bwb#: OWL2 Profile
  No violations found in profile.
  No violations found in profile.
HermIT reasoner preparing...
  Building the class hierarchy...
    ... finished
  Classifying object properties...
    ... finished
  Classifying data properties...
    ... finished
Consistent: true
The following classes are unsatisfiable:
  <bwb:BWB_GM_DV_QUANTITY>
```

Example 6. Validating Archetype BODY_WEIGHT_BIRTH against BODY_WEIGHT and RM

Including the BODY_WEIGHT_BIRTH archetype does not affect the property of the combined ontology as belonging to the OWL DL profile. Furthermore, the combined ontology is still consistent. However, now there is one class in the ontology, BWB_GM_DV_QUANTITY that is ruled out to be unsatisfiable. This means that from a logical point of view, this class cannot possibly have any individuals, it is inconsistent with some axioms of the ontology. Since this class is part of the BODY_WEIGHT_BIRTH archetype, we can conclude that this archetype itself is invalid with respect either to the RM or to the archetype BODY_WEIGHT, or both.

The reason for this inconsistency turns out to be trivial (which is usually not the case when it comes to validation of large ontologies). The class BWB_GM_DV_QUANTITY is defined to be a subclass of BW_LB_DV_QUANTITY, which has a restriction on a data type property DV_QUANTITY-units, stating that it must have the value "lb". The subclass states that the same data type property must have the value "gm". There can exist no individual such that the same single valued property has both the value "lb" and "gm", so the class is unsatisfiable.

This can be corrected either by altering the BW archetype or the BWB archetype:
• Altering the BW archetype:
  by stating that the BW_LB_DV_QUANTITY has a data type property DV_QUANTITY-units restricted to the value "gm"

• Altering the BWB archetype:
  by stating that the BWB_GM_DV_QUANTITY has a data type property DV_QUANTITY-units restricted to the value "lb"

Either way, the archetype BODY_WEIGHT_BIRTH validates without problem against the RM and against BODY_WEIGHT. The potential and the limitations of automated reasoning will be discussed in the next section.
7. Discussion

It is the tension between creativity and skepticism that has produced the stunning and unexpected findings of science (Carl Sagan)

In this section, I will first summarise the results and observations before discussing validity, generality and limitations.

7.1. Results and Observations

7.1.1. Results

• The OpenEHR Reference Model was implemented as an automatic transformation from the XML-Schema to OWL/XML. Since other projects have automated the second part, i.e. the transformation of archetypes into OWL, the door is open to an automatic and maintainable transformation of the entire OpenEHR Health Record (RM+archetypes) into OWL. Future work in this direction (cf. Section 8.1, “Completing the Archetype Ontologising process” and Section 8.2, “Formalisation of the Semantic Distillery”) is suggested.

• The mapping from XML-Schema to OWL was performed with relative ease and minor issues only (Section 6.3.4, “Transformation Outline” Section 6.3.5, “Mapping Details”)

• A couple of archetypes were manually translated into OWL as restrictions of concepts in the RM

• Some trivial "proof of concept" validations were performed, using a formal reasoner, with expected results

7.1.2. Some Observations

• Implementing the entire battery of restrictions in the RM led to very long (approx. 2 hours) classification time for the reasoner. Removing one type of restriction (with some loss of semantics) the classification time was greatly reduced. Furthermore, the reasoner did not accept one type of restriction, although it was classified as valid OWL by the OWL API.

Future work is suggested to investigate these issues (Section 8.4, “Optimising Reasoner Performance”)

• The manual mapping process of the chosen archetypes revealed some errors. Section 6.5, “Some issues”. The underlying problems would have to be addressed when performing an automatic transformation from ADL to OWL, thus this process in itself can catch some errors and serve as a quality enhancing mechanism

7.2. EHR Terminology Binding — the Code Binding Interface

The guiding hypothesis (cf. Section 2.3.2, “Working Hypothesis”) for this work has been that a common semantics based representation, such as OWL is suited or maybe even needed for comparing the semantics of the EHR systems and their interrelationship with a terminology system.

It has been beyond the scope of this work to get an answer verifying or falsifying the working hypothesis, although some ground hopefully has been prepared for it. Nevertheless a brief discussion of a possible procedure will follow, in order to outline future work along this line.
A recapitulation of the postulated preconditions for the usage of OWL as a bridging technology for validation and reasoning over semantic compatibility issues are the following:

- The EHR systems have an OWL based representation
- The terminology system is expressed in OWL
- There is a method for binding the Terminology to the EHR, i.e. an interface between the EHR and the Terminology also expressible in OWL.

This work has aimed at realizing the first point above. Using SNOMED CT, the second condition holds (cf. Section 3.1.3, “SNOMED CT”). Now, the last assumption must be verified: We need some method for combining the “EHR ontology” with the Terminology. Alan Rector has described such a method, and dubbed it the "Code Binding Interface" (CBI) [RECTOR07], [RECTOR09]. All references to the Terminology System will be made indirectly through the CBI. The CBI serves the purpose of “gluing” together the constituent sub ontologies so that these can be merged into one combined ontology. The realization of a CBI would be the next logical step in the continuation of this work (cf. Section 8.3, “Work on the CBI Concept”).

7.3. ADL versus OWL

When comparing and reasoning about the different formalisms that are available for representing EHR related knowledge and data, it is important to be aware of the different abstraction levels, in order to understand what formalism is best suited to a particular purpose.

In [ADLBOOK07] there is a listing of the merits and shortcomings of ADL and OWL respectively, with a certain bias in favour of ADL for representing EHR data. As an example, the lack of XPath-like navigation is listed as a lacking feature of OWL, which in contrast is present in ADL. In my opinion, this list incites misleading associations and even comes to a wrong conclusion due to the lack of keeping track of and emphasising the different abstraction levels of the two languages, which determines when their respective use is favourable. Quoting from the concluding section of the comparison:

To use the archetype on data, the data themselves would have to be converted to OWL, i.e. be expressed as ‘individuals’. In conclusion, we can say that mathematical equivalence between OWL and ADL is probably provable. However, it is clear that OWL is far from a convenient formalism to express archetypes, or to use them for modelling or reasoning against data.

It is probably the case that OWL would not be a suitable representation for storing the EHR-data itself. It should be seen as a mechanism for representing the schema structures, (such as archetype definitions) for reasoning and validation purposes. Furthermore, mathematical equivalence between OWL and ADL is not provable, because they express different things, albeit with a certain overlap. OWL is designed on a mathematical base specifically for the purpose of reasoning and is superior in this respect. That might also make it a strong candidate for modelling of archetypes. Whichever is most suitable for presenting the archetype to a human for reading or manipulation is another matter, and is the job of an appropriate tool.

As far as I can see, the OpenEHR approach of having a generic Reference Model that is constrained by ADL to create categories of con-formant hierarchical data structures has resemblances to XML being constrained by an XML-Schema (or by Relax NG). The precursor to XML, SGML which was developed during the seventies and standardised by ISO in 1986, already had this mechanism through its Document Type Description (DTD) mechanism. The DTD was the ancestor of XML-Schema, and used even in XML before being superseded by XML-Schema (or Relax NG). So one variation of the two level approach to describe and constrain hierarchical data structures has a long history (relative to the short history of computer science).

The difference between OpenEHR RM/ADL and XML/XML-schema are mainly

- The OpenEHR RM has a "domain bias", and is therefore more extensive and expressive than XML which is a minimal generic model to describe hierarchical textual structures. This means that more
has to be expressed in an XML-Schema in order to obtain a document class that is semantically equivalent to an archetype.

- The OpenEHR RM and Archetype Object Models are modelled in UML, and there is a direct correspondence between an ADL instance and a run time instantiation of the AOM. These models have an object oriented bias, which is beneficial when seen as representations for software systems, where they may be further equipped with logical artefacts, such as pre- and postconditions, loop and class invariants and finally operational implementation artefacts.

Therefore, the XML/XML-Schema models can be considered more abstract than the OpenEHR RM/AOM/ADL models, in the sense that the latter are more "implementation oriented". OWL, on the other hand, is even more abstract, omitting as much as possible about internal structures and reducing the description to a set of statements about what is true in the description. But this minimalism is what enables formal reasoner engines to operate on the models, assisting us in discovering irregularities and inconsistencies. In this respect "less is more" — being more restrictive in the "decoration" of the models allows for formal reasoning services, which according to the working hypothesis are supposed to be valuable if not even necessary tools in the quest for semantic interoperability.

### 7.4. Ontology and Ontology Alignment Issues

We have discussed the need for establishing some common semantic framework between the EHR systems and the Terminology. The question is whether this is enough. In [CEUSTERS03] a case is made that Description Logics are not enough to reach the goal of semantic interoperability between the EHRs and the medical terminology system. The ontology itself has to be built upon certain ontological principles and one of the most important is that the ontology should describe the reality as faithfully as possible.

According to [BODENREIDER04], sound terminology design also implies keeping a clear distinction between Ontology — the study of the description of reality itself and Epistemology — the study of matters related to knowledge, such as how knowledge is acquired, the nature of knowledge etc. The authors show that several studied biomedical terminologies contain terms that denote features having different and incompatible qualities. In some cases they represent invariant features representing the reality, and thus correctly belong in the domain of ontology. In other cases, they carry information on how this reality is captured or understood by health care professionals, thus, information not belonging in the domain of ontology, but rather in that of epistemology. The authors argue that such inconsistencies defy the purpose of medical terminologies.

The case for the realist view of ontology when designing medical terminology systems as well as EHR systems is made in [CEUSTERS06]. The authors argue that the mathematical (model theoretical) principles underlying Description Logics are too "neutral" to provide any guidance in the design of ontology systems. Therefore, realist based ontology engineering is necessary. Also, in order to obtain some coherence between the ontologies at the EHR and terminology levels, the introduction of a series of "top level" ontological categories are needed.

In the process of sorting out the issues relating to the "boundary problem" as described in Section 3.3.2, "The Interface between EHRs and Terminology Systems", there is a need to strike a balance between what semantics is represented in the EHR, and what is in the Terminology system. In [RECTOR08] the case is made for increasing the semantic expressiveness of SNOMED CT, and thus overcoming some inherent problems when expressing some constructs in the terminology. A natural consequence of increased expressiveness on the terminology side would be to try to further eliminate implicit semantics regarding the same concepts on the EHR side. This leads naturally to the question whether there is a need for a series of "reference ontologies" for the EHR. Instead of "reverse engineering" the semantics of the EHRs in a "semantic distillation" process, maybe the "reference ontologies" could be a basis for the specification of the EHR systems.
8. Future Work

The best thing about the future is that it comes only one day at a time (Abraham Lincoln)

8.1. Completing the Archetype Ontologising process

As previously stated, an automatic transformation of archetypes from ADL to OWL has been implemented [LEZCANO08], based on a hand coded version of the OpenEHR RM [ROMAN04]. Using this work as a base for an automatic archetype generation, based on OWL 2 and the auto generated RM from the present project (or some refined successor of it) would be a natural follow-up, potentially enabling full automation of the OpenEHR to OWL 2 transformation.

8.2. Formalisation of the Semantic Distillery

Work should be done to validate the "semantic distillation" process in itself. This applies both to the auto generated RM, which is created by a rather ad hoc process, and to the potentially auto generated archetypes.

This raises the question of "validating against what", the possible need for some reference EHR ontology, ontology alignment issues (cf. Section 7.4, "Ontology and Ontology Alignment Issues", calling for more theoretical studies and the usage of Formal Methods.

8.3. Work on the CBI Concept

The CBI concept was briefly discussed in Section 7.2, “EHR Terminology Binding — the Code Binding Interface”. When all subsystems are have an OWL representation, this could be implemented. Of special interest would be to see how experiments with variations of the CBI could be analysed and validated by a reasoner applied to the combined ontologies. This might give insight into the "boundary problem" (cf. Section 3.3.2, “The Interface between EHRs and Terminology Systems”) and related problems.

8.4. Optimising Reasoner Performance

The work was done with a beta version of the OWL API 3.0. and the availability of reasoners for this re write of the api was limited to the HermIT reasoner. Although it is supposed to be very fast, it took a long time (two hours) to classify the RM when all constructs were used. Therefore we used a simplified version of the transformation in order to have manageable reasoning times. It is a well known that the reasoning algorithms have a complexity that sometime lead to unacceptable classification times, and much work is done in this area in order to improve their performance. But sometimes good results can be achieved by more simple tricks of the trade, such as reformulating the constraints, change to another reasoner, more optimised for the problem at hand, or simply increasing the memory etc. In order to make the method realistic, it is necessary to have reasoner times which allow for extensive experimentation, which implies running the reasoner over and over again.

8.5. The EHR from an MDA Perspective

MDA is as the name suggests, more about architecture and software engineering methods than about ontologies and formal methods, although research is done about applying formal methods in the MDA development process. But the need for combining MDA with ontology oriented methods are acknowledged, as demonstrated by the release of the ODM which is a family of meta models aimed at ontology engineering (cf. Section 4.5.2, “Model Transformations”). It is early to say whether the combination of OWL MDA will become wide spread, but at least, the link is established.
Interesting work in this direction would be to work with the accommodation of EHR models within the M3 Model framework, and to investigate possible benefits from using model transformations “à la MDA” to build the “semantic distillery”.

Another potential use case where the MDA/Model Transformation Approach could have some benefits is in so called “round trip engineering”. The “semantic distillery” transforms the models into a form that is not suitable for the run time EHR. If these models are changed or corrected in the OWL representation, then there is a need for a reverse transformation, capturing these changes. Possibly, this type of problems are suitable within an MDA framework.
9. Conclusion

Our achievements of today
are but the sum total of our thoughts of yesterday.
You are today where the thoughts of yesterday have brought you
and you will be tomorrow where the thoughts of today take you
(Blaise Pascal)

Do the proposed methods and tools lead us into a fast-track to the holy grail of a semantically inter-connected Health Record, or are we venturing into the domain of “the alchemy of semantics”? Most probably not the first and hopefully not the latter!

One aim of the present work has been to contribute to increased awareness of semantic methods and technologies when trying to cope with problems relating to semantic interoperability of distributed EHRs. A central point has been making the case for the usage of formal semantic methods in order to be able to making authoritative statements about semantic issues, such as semantic interoperability. Furthermore, some work has been done to show the practical feasibility of such an approach.

Even if we should succeed in transforming a set of Health Records to a semantic representation there remains the complicated problem of aligning the ontologies. I do not think that this problem is solved only by creating a terminology binding for each system to a common terminology, because there are semantic alignment problems between the health records themselves as well. Maybe a set of common Reference EHR Ontologies could be a step in the right direction. The result of the “semantic distillery” process of each system could then be independently validated against the common Reference EHR Ontology.

The problem of semantic interoperability is hard, deep and vast. On one hand, it does not go away simply by ignoring it. On the other hand, some solution to it must be found in order to be able to make good use of modern information and communication technology for the medical disciplines. Such support has a potential to increase their asset in the fundamental areas of information and of knowledge.

The problems of are likely to be solved gradually and as a result of focussed research. Rising the awareness of the necessity of applying formal semantic methods when dealing with problems that are semantic in essence will catalyze that process.
## Glossary

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOM</td>
<td>Archetype Object Model [<a href="http://www.openehr.org/releases/1.0.1/architecture/am/aom.pdf">http://www.openehr.org/releases/1.0.1/architecture/am/aom.pdf</a>]. A generic object model for expressing constraints on a RM in order to implement a clinical concept (an archetype).</td>
</tr>
<tr>
<td>CBI</td>
<td>Code Binding Interface</td>
</tr>
<tr>
<td>CEN</td>
<td>Comité Européen de Normalization [<a href="http://www.cen.eu/cenorm/homepage.htm">http://www.cen.eu/cenorm/homepage.htm</a>]. (European Committee for Standardisation) Official (EU-sponsored) European Standardisation body, coordinating the standardisation work of 30 National Members in order to develop voluntary European Standards (ENs). It organises different work groups, and CEN/TC 251 (CEN Technical Committee 251) is a work group within the European Union working on standardisation in the field of Health Information and Communications Technology (ICT) in the European Union. The EN 13606 is a result of the work of CEN/TC 251.</td>
</tr>
<tr>
<td>DL</td>
<td>Description Logics [<a href="http://dl.kr.org/">http://dl.kr.org/</a>]</td>
</tr>
<tr>
<td>EHR</td>
<td>Electronic Health Record (cf. Section 3.2, “The Electronic Health Record”) The persistent longitudinal record of health and care information about a single subject of care (patient).</td>
</tr>
<tr>
<td>EMF</td>
<td>Eclipse Modelling Framework [<a href="http://www.eclipse.org/modeling/emf/">http://www.eclipse.org/modeling/emf/</a>] An Eclipse foundation MDE project, strongly influenced by MDA.</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
  The ICD is currently in its tenth edition, ICD-10, which is a coding of diseases and signs, symptoms, abnormal findings, complaints, social circumstances and external causes of injury or diseases, as classified by the World Health Organisation. The code set allows more than 155,000 different codes. |
| IDE     | Integrated Development Environment  
  Programming tool aimed at enhancing programming productivity by tying together other tools in the programming tool chain such as syntax sensitive program editors, compilers and debuggers. |
| IHTSDO  | The International Health care Terminology Standards Development Organisation [http://www.ihtsdo.org]  
  Non-profit organisation with 13 member states (of which Sweden is one) for the development and maintenance of health terminology systems, in particular SNOMED CT. |
  Internet standard defining a new protocol element as a complement to the Uniform Resource Identifier (URI). An IRI is a sequence of characters from the Universal Character Set. A mapping from IRIs to URIs is defined, which means that IRIs can be used instead of URIs, where appropriate, to identify resources. |
  ISO organises a number of Technical Committees (TCs) for the different standardisation domains, and the ISO/TC 215 is working on the standardisation of Health Information and Communications Technology to enhance interoperability between systems. |
| MDA     | Model Driven Architecture [http://www.omg.org/mda/]  
  One of the prominent standards for Model Driven Engineering MDE, hosted by OMG. |
| MDE     | Model Driven Engineering is a software development method focussing on the development of models from the abstract to the more concrete in a refinement process driven by model transformations. The method can be formalised to the extent that these model transformations can be formalised. MDA and EMF are prominent MDE initiatives. |
  MOF is a three layer architecture for defining models and meta models. Therefore it is often called an M3-model. It is often called a four layer architecture, but the last level is the “system” and hence not really part of the models. |
| ODM     | Ontology Definition Meta-model [http://www.omg.org/spec/ODM/1.0/] |
A specification establishing the formal grounding for representation, management, interoperability, and application of business semantics for Model Driven Architecture (MDA) based software. It is the basis for a family of specifications that marry MDA and Semantic Web technologies to support semantic web services, ontology and policy-based communications and interoperability, and declarative, policy-based applications in general.

**OMG**

An industry consortium aimed at setting standards for modelling and distributed object oriented technology. The de facto standard for OO design, UML is managed by OMG, as well as the MDA initiative.

**OpenEHR**
OpenEHR [http://www.openehr.org]

cf. Section 3.2.1, “The OpenEHR Foundation”.

**OWL**
Web Ontology Language [http://www.w3.org/TR/owl-features/]

The language was originally built upon the foundations of RDF and RDFS, but it had also DL antecedents, and was right from the outset defined according to a strict formal semantics, adhering to the principles and theory of Description Logics. The language is developed and maintained by W3C, and is designed to be compatible with the distributed architecture of the Web. There exist a number of open source reasoners and implemented systems supporting it.

**OWL 2**
OWL 2 Web Ontology Language [http://www.w3.org/TR/owl2-syntax/]

The OWL 2 Web Ontology Language (informally OWL 2) is an extension of the W3C OWL Web Ontology Language. The new features include additional property and qualified cardinality constructors, extended data type support, extended annotations and the definition of three profiles (subsets) of the language, optimised for particular usage scenarios, as well a new XML-friendly serialisation syntax, OWL/XML.

**RDF**
Resource Description Framework [http://www.w3.org/TR/rdf-primer/]

The Resource Description Framework (RDF) is a language for representing information about resources in the World Wide Web. RDF is maintained by W3C

**RDFS**
RDF Schema [http://www.w3.org/TR/rdf-schema/]

RDF Schema is a specification that describes how to use RDF to describe RDF vocabularies and defines a vocabulary for this purpose. RDF Schema is maintained by W3C

**RM**
Reference Model

Term used by OpenEHR to denote the information model of the health record. Clinical concepts are by constraining the reference model by means of archetype definitions, specified with ADL and
instantiations of these concepts (i.e. actual clinical data) are implemented according to AOM.

**SGML**


A precursor to XML, introducing the idea of describing electronic documents (general text representable data) by applying a Document Type Definition (DTD) to a hierarchical textual structure. XML is a simplified successor to SGML, retaining most of its features.

**SNOMED CT**

Systematised Nomenclature of Medicine - Clinical Terms [http://www.ihtsdo.org/snomed-ct/]

A coding system, controlled vocabulary, classification system and thesaurus owned and maintained by IHTSDO. Several national language translations projects of the vocabulary are currently underway one of which is Swedish.

**SWRL**

Semantic Web Rule Language [http://www.w3.org/Submission/SWRL/]

A Rule Language Combining OWL and RuleML (Rule Markup Language).

**UML**

Unified Modelling Language [http://www.uml.org/]

UML is a general-purpose modelling language in the field of software engineering. It is particularly aimed at object oriented design, modelling specification and documentation. The standard is managed by the Object Management Group (OMG).

**URI**

Uniform Resource Identifier [http://www.ietf.org/rfc/rfc3986.txt]

A Uniform Resource Identifier is a compact sequence of characters that identifies an abstract or physical resource. The URI syntax defines a grammar that is a superset of all valid URIs, allowing for different schemes for URIs.

**W3C**

World Wide Web Consortium [http://www.w3.org]

The World Wide Web Consortium (W3C) is an international community that develops open Web standards to ensure the long-term growth of the Web. The consortium maintains the central standards driving the web, such as HTTP, HTML, and also those defining the Semantic Web [http://www.w3.org/2001/sw/].
Bibliography


Appendix A. Open Source Tool Set

The following is a comprehensive list of the open source software tools used for the practical/experimen-
tal part of the report as well as for writing, formatting and styling it.

1. XML Transformation Tools and Libraries

Semantic Web technologies, being web-centered by design, rely heavily on XML in their suite of
standards specifications. Therefore, we need tools for editing, parsing, transforming and validating
XML.

**Emacs/nXML-mode: Schema aware XML Editor**

GNU-Emacs is a powerful, flexible and versatile text editor. It is customisable via an extension mechanism. Various add-ons exist, recognising different types of syntax. Emacs can be directed to recognise the type of file being edited and automatically activate a corresponding syntax sensitive editing "mode". There exists an excellent XML-mode for Emacs, called nXML mode. It uses Relax NG as its schema language, and not XML-schema, but this is not a significant restriction, because Relax NG is a very powerful schema language standardised by the not-for-profit consortium OASIS. It is gaining increasing traction by XML users, and also, there exists tools for conversion from XML-schema to Relax NG schemata.

**Xsltproc: XML Transformations with XSLT.** The OpenEHR Reference Model specifications expressed in XML-Schema were transformed into the OWL/XML serialisation format by means of XSLT (cf. Section 6.3, “Mapping of the OpenEHR RM to OWL”), a language which is designed for the purpose of transformation from one representation of XML into another. The command line tool xsltproc is an XSLT processor which was used.

**XML Schema Validators.** After conversion of the Rm Schemata to OWL 2/XML, there is a need to check that the transformed result actually conforms to the OWL2 XML Serialisation Schema. For this, I use the Schema tools from "Basic Research in Computer Science" (BRICS), a Research Institute within the areas of computer and information sciences, hosted by the Universities of Aarhus and Aalborg in Denmark.

**Java OSS Libraries.** Both xsltproc and the BRICS schema tools depend on the XML parser SAX-Parser which is distributed as part of the standard Java libraries in the Java environment.

2. OpenEHR Related Tools and Libraries

**OpenEHR LIU Archetype Editor.** This tool ([FORSS06]) was used for browsing and inspecting the Archetypes involved. Initially a Master project, IMT/LIU has continued the development of this Archetype Editor implemented in Java, and based on the Java RM implementation.

**OpenEHR RM, Java Implementation.** This implementation of the OpenEHR Reference Model is at the core of any Java based EHR based on the OpenEHR specifications, such as e.g the LIU Archetype Editor.

**OpenEHR ADL-parser, Java Implementation.** This tool is logically a part of the Archetype Object Model specifications, but comes packaged with the Java RM implementation. It is also an essential base component, used by the Archetype Editor.
3. Semantic Web Related Tools and Libraries

Protégé-OWL\textsuperscript{11}. A Java based, open source OWL editor. Protégé features a plug-in architecture for incorporating extensions, and provides an Application Programming Interface (API) for integration with other knowledge-based tools and applications.

OWL API\textsuperscript{12}. A Java based API for working with OWL ontologies, implementing the new OWL 2 language. Its design and features are outlined in "The OWL API: A Java API for Working with OWL 2 Ontologies" [HORRIDGE09]. Protégé 4 is built upon this API. The version 3.0 was released in February 2010. Before that a development snapshot of the API was used.

HermiT\textsuperscript{13}. An efficient DL reasoner for OWL, compatible with the OWL API. The HermiT reasoner was the only one that at the time of the writing was compatible with the new OWL API 3.0. Other well known reasoners that will also soon be available for the new API are Fact++\textsuperscript{14} and Pellet\textsuperscript{15}.

4. Program Editor and Version Control System

The IDE Eclipse\textsuperscript{16} was used for Java editing and debugging, and Subversion\textsuperscript{17} was used as version control system throughout the project.

5. Publishing Tools and Libraries

The publishing method used was DocBook/XML. This enables the usage of XML tools for transformation to diverse output formats, while not sacrificing good control of the output styling. The following figure gives an overview of the "publishing pipeline", further detailed below.

![Publishing Tool Pipeline Diagram](image)

**Figure A.1. Publishing Tool Pipeline**

**XML Editing with Emacs.** The source format of the present report is XML, using the DocBook 4.5\textsuperscript{18} Schema, the text editor being Emacs/nXML-mode, (cf Appendix A, Section 1, “XML Transformation Tools and Libraries”.

**DocBook XSL Stylesheet Distribution.** The report was translated from DocBook/XML to HTML and to XSL-FO\textsuperscript{19} for further processing to PDF. This processing was performed with an XSLT processor (see below) using the standard (but customised) DocBook XSL Stylesheets.

**Saxon: XSLT Transformations.** The XML source is transformed to HTML and PDF using XSLT. In addition to the processor xsltproc (Appendix A, Section 1, “XML Transformation Tools and Libraries”)

\textsuperscript{11} http://protege.stanford.edu/
\textsuperscript{12} http://owlapi.sourceforge.net/index.html
\textsuperscript{13} http://hermit-reasoner.com/
\textsuperscript{14} http://code.google.com/p/factplusplus/
\textsuperscript{15} http://clarkparsia.com/pellet
\textsuperscript{16} http://eclipse.org
\textsuperscript{17} http://subversion.tigris.org/
\textsuperscript{18} http://www.docbook.org/specs/docbook-4.5-spec.html
\textsuperscript{19} http://www.w3.org/TR/xsl/
The XSLT processor Saxon\(^{20}\) was used. The reason for using two different processors is that one was considerably faster (xsltproc), while the other had more functionality (Saxon). The first one was used in the tight edit/generate/view loop, while the other one was used from time to time and for the final result in order to generate output with some extra visual features.

**Apache FOP: Print Formatting with XSL Processor.** The XSL-FO to PDF processing was performed with the Apache FOP\(^{21}\) tool. This framework makes use of the Apache Batik SVG toolkit\(^{22}\) for rendering the SVG images.

**SVG editor.** The figures are created using Vector Graphics\(^{23}\) for resolution independent output to various other image formats.

The Vector Graphics Editor *Inkscape*\(^{24}\) was used as SVG-editor.

## Appendix B. Lab Artefacts

The various components resulting from the practical work are located at a directory reachable by the following URI:

http://inherit.se/semweb_ehr/

The file README.txt [http://inherit.se/semweb_ehr/README.txt], located at the root of this directory describes the directory structure and contents, and how to configure and run the tools (as well as the necessary prerequisites for doing so, such as program and library dependencies). At this location there is also a zip-archive, semweb_openehr.zip, containing the entire directory structure.

The most important components of this directory are the following:

- **OwlConv** Test program using the OWL API v3.0 for validation and transformation to other OWL syntaxes.
- **xsd2owl** XSLT program for transformation of OpenEHR RM to OWL/XML

---

\(^{20}\) [http://saxon.sourceforge.net/]
\(^{21}\) [http://xmlgraphics.apache.org/fop/]
\(^{22}\) [http://xmlgraphics.apache.org/batik/]
\(^{23}\) [http://www.w3.org/Graphics/SVG/]
\(^{24}\) [http://www.inkscape.org/]