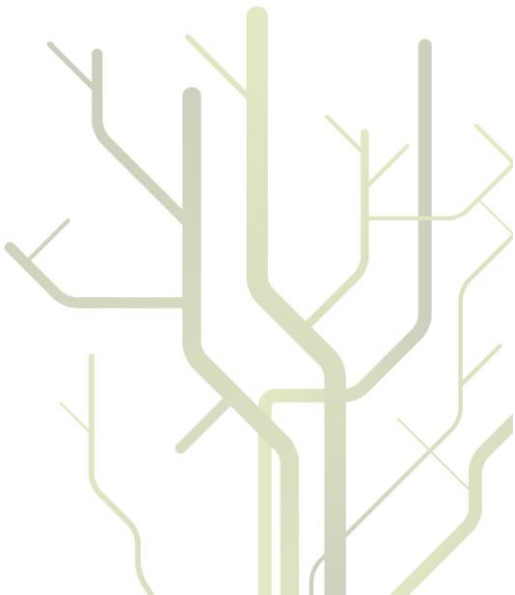
UNIVERSITY OF TROMSØ UIT

FACULTY OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE

Enabling medical research on clinically collected data using openEHR archetypes

Leykun Melkamu Gebeyehu

INF-3997
Master's Thesis in Telemedicine and e-health
May 2012



Abstract

Purpose: The aim of the thesis is to provide a high quality research repository based on openEHR platform and assess the feasibility of the approach. This involves transforming the extracted data to openEHR data format, annotating, and visualizing the committed data.

Motivation: The main source of motivation was article 6 of the Helsinki declaration which stresses the strict requirement on continuously challenging the method used in healthcare. This project enables building a system that can be used to continuously assess the outcome of procedures and guidelines used in healthcare.

Method: We designed the application using the design paradigm rooted in the engineering discipline and the feasibility of the system was assessed using a criterion. Using this approach, we iterated through a requirement specification and design until a reliable, maintainable design that fulfils the expectation was achieved. After building the system, the feasibility of the solution was assessed using a previously developed criterion. The criterion use the effect of the solution and the effort needed to build the system to assess the feasibility. We assigned value to the variables in the criterion using evidence collected form the application building process and the application itself.

Result: We mapped the extracted data to openEHR platform using flat archetype which mimic the structure of legacy data. We queried back this data and transformed it to another composition instance based on design archetypes which can be used as a gateway to annotate data. The data in the repository was visualized using windows form application.

Conclusion: The overall feasibility of developing a research repository using openEHR platform

showed high potential effect, but also medium effort attached to it. The system provides a standardized and structured high quality data. However, developing the system in addition to internal resource needs a strong collaboration with healthcare workers. The process of annotating and expanding the data using archetypes could also need some time, but this approach is easier when compared with traditional methods used to annotate data.

Preface

This work is intended to developers and researchers who are working in the field of openEHR applications. This thesis is done as a part of TILTAK 41 national project to test the feasibility of applying archetypes to research repositories. The project focus on reusing existing archetypes provided by openEHR, and designing new archetypes when needed.

It is known that Secondary use of electronic health record data for research can improve health-care experiences for individuals . However, there has been little research to support these claims. Though there were some systems built to enable clinical research from large health information collections, they have been criticized since the quality of the clinical data were not up to research standards. The aim of this thesis is to fill some of this gap by enabling research on data collected from electronic health record using openEHR as a platform of the solution. In this study, we assessed the feasibility of the approach.

I am pleased to acknowledge the people who assisted me for completing this thesis. First and foremost, my deepest gratitude goes to my supervisor associate professor Johan Gustav Bellika for tirelessly working with me from the beginning up to the end and sharing me his broad knowledge and experience. He has been following up the progress of my thesis with great care and commitment. Our discussions taught me a lot beyond the thesis. He have spent a great deal of his valuable time reading and commenting each and every page. He earn my gratitude more than anybody else.

Special thanks to the Norwegian state educational loan fund-Lånekassen for the scholarship grant during my study at the university of Tromsø.

I appreciate DIPS's for the materials, schema, and technical help they provided me while extracting the data. I would also like to thank Ronny Bergeton Thomassen for his kind assistance so that I could work in collaboration with DIPS. My profound gratitude goes out to Håvard Ballo for his valuable help on extracting the data from DIPS electronic health record. Trond Elde for all the help he provided me while I was analysing the graphical user interface used by the healthcare workers.

I would like to acknowledge ocean informatics for providing me with the platform and technical assistance needed to interact with the server.

I would like to thank Knut Magne Augestad and Femi Oyeyemi for their valuable inputs while specifying the requirements of the system.

Most importantly I thank the eternal, most blessed God, creator of the universe, who fathoms all contexts. What would I do without Him!

Last but not least, I would like to thank my family members and friends who helped me in doing this project, Desalegn and Temesgen for their wonderful help on LaTeX .

Contents

List of Figures

List of Tables

Abbreviations

EHR	Electronic Health Record
CEN	European Committee for Standardisation
HL7	Health Level Seven
RIM	Reference Information Model
RM	Reference Model
AM	Archetype Model
SM	Service Model
API	Application Programming Interface
CKM	Clinical Knowledge Manager
AQL	Archetype Query Lanaguage
ADL	Archetype Definition Language
AOM	Archetype Object Model

TDS	Template Data Schema
XSLT	EXtensible Stylesheet Language
SNOMED CT	Systematized Nomenclature of Medicine–Clinical Terms
ETL	Extract Transform and Load
CSV	Comma-Separated Values
XML	Extensible Markup Language
HTML	HyperText Markup Language
ICD	International Classification of Diseases
XSD	XML Schema Definition
FDA	Food and Drug Administration

Chapter 1

Introduction

1.1 Background and motivation

An increasing number of studies are stating the value of the electronic health record(EHR) for improving clinical care through reminders, alerts, and other forms of clinical decision support system. However, it is known that lifelong electronic health records can also supply valuable information for research and quality management in addition to supporting treatment of patients (Hersh 2007), (Wrba et al. 2008) (Van der Lei 2002).

Secondary use of electronic health record data for research can improve healthcare experiences for individuals, expand knowledge about disease and appropriate treatments, and strengthen our understanding about effectiveness and efficiency of health care systems (Safran et al. 2007).

Until now, there has been little research to support these claims (Hersh 2007) (Wrba et al. 2008). Even if there were some systems built to enable clinical research from large health information collections, they have been criticized since the quality of the clinical data were not up to research standards (Weiner & Embi 2009).

The goal of this thesis is to fill this gap by exploring possible effective method of using electronic health record data to support not only clinical practice but also research, quality assurance, and education to facilitate continuous flow of information and knowledge in the healthcare domain. This thesis is done as a part of TILTAK 41 national project to test the feasibility of applying archetypes to research repositories in Norway.

The first source of inspiration to this thesis was article 6 of declaration of Helsinki which states that (Touitou et al. 2004):

"*The primary purpose of medical research involving human subjects is to improve prophylactic, diagnostic and therapeutic procedures and the understanding of the aetiology and pathogenesis of disease. Even the best proven prophylactic, diagnostic, and therapeutic methods must continuously be challenged through research for their effectiveness, efficiency, accessibility and quality*".

The article stress the strict requirement on continuously assessing the methods used in healthcare and this thesis provides a solution to this problem by developing a system that empowers researchers to continuously challenge and investigate the methods used in healthcare.

The other driver towards the thesis was the complexity and rapid development of medicine. According to Alan Rector knowledge in healthcare is constantly changing in three ways: In breadth, because new knowledge is always being discovered or becoming relevant; in depth, because finer-grained detail is always being discovered or becoming relevant and in complexity, because new relationships are always being discovered or becoming relevant (Rector et al. 1999). As a result of rapid change in healthcare knowledge, treatment procedures and guidelines used in healthcare can quickly become outdated. The project enables evaluating the outcome of procedures and guidelines used in healthcare. This will help to identify potentially outdated methods and those methods can be further investigated if they are supported with current knowledge. In this way the developed

system can help to cope with the enormous growth in medical knowledge.

Outcome assessment in healthcare is performed with the goal of improving the quality of care. Recently more attention is being devoted to it and this demands the data collected during documentation of clinical work to be transformed and standardized so that it can be easily available for outcome assessment and research. Since the project enables controlling the quality of care by allowing outcome assessment and research on standardized data, the current focus on quality of care implies that the thesis can have an impact on the future healthcare system. As a result, it was another motivation towards the thesis (Chassin 1996).

1.2 Scope and research problem statement

The research problem can be summarized in the following statement:

"*Is it feasible to use openEHR to make the data in the routinely collected DIPS electronic health record available for research and outcome assessment*?"

The data source in this study is DIPS electronic health record which is organized as a relational database. It contains data in structured and unstructured format which makes the process of computing and analysing the data for research purpose more difficult.

The project aim at providing a high quality research repository for the extracted data from the DIPS Electronic health record so as to enable assessment of the quality of the medical services through research. It allows us to continuously challenge and investigate the methods practiced in routine activities in healthcare.

There are different methods that can be used for storing and presenting research data collected from electronic health record. For example, one can use OpenQReg or openEHR to store healthcare data. Studies claim that openEHR has the potential to build re-

search repositories (Chen 2009) and it is a promising approach for achieving semantic interoperability between heterogeneous health information systems. However, openEHR implementations are not in wide use in healthcare practice and their applicability remains the subject of research (Anani et al. 2011) . Our target is to prove the claim that openEHR can be used as a platform to build research repository and assess the feasibility of the solution.

Collecting, organizing, presenting, and disseminating healthcare knowledge is a wide scoped challenging process. In general interconnecting bioinformatics systems is a complex task. It is well known that complex ethical, political, technical, and social issues surround the secondary use of healthcare data. Generally when developing a research repository, the researchers start with patient consent and ethical board approval. Then the developed privacy policies and security solution are evaluated to check if the system is ethical and safe to use (Safran et al. 2007).

However, in this thesis ethical, political, and social issues are not studied, we are using pseudo patient data that would not put patient privacy at risk. Our dataset is also limited to data related with diagnosis of malignant neoplasm of colon International Classification of Diseases (ICD-10) code of C!(). The project is focused on studying the feasibility of the proposed solution by analysing the technical issues; effort needed and expected value gained of using the system.

In this study, we are extracting only the structured data from the DIPS electronic health record. The first task while building the research repository is converting this data to openEHR format. The result of this process is data that conforms to openEHR reference model data format. The next task is giving meaning to the data that has been imported to openEHR platform. This is accomplished with the archetypes that are created by the clinicians themselves.

1.3 Who is in this area?

To explore the current status of openEHR and to find related works, we did a literature review from ScienceDirect, IEEE Explore, Pubmed, Google Scholar, ACM Digital library and other libraries.

Among the related works was a study, which has proved that the archetype format is more expressive and can be used to preserve legacy EHR content definitions. It states that archetypes can facilitate interoperability between systems in order to allow cyclic flow of information and knowledge among the partition care, research and education to achieve a safer and more efficient healthcare system (Chen 2009). This study also specifies that openEHR has a potential to build a research repository and it should be further researched. We were also able to find a number of studies in the field of semantic interoperability and archetypes, and most of the papers stated the importance of archetypes for semantic interoperability between electronic health records (Wollersheim et al. 2009).

A closely related study explored the feasibility of using the openEHR approach to support multi-centre research in comparison to a system called extensible architecture for using routine data for additional purposes (eardap). The study revealed that openEHR is suitable to enable multi-centre clinical research on data collected form healthcare practice (Garde et al. 2005).

To our knowledge there have been no other studies, which examined the feasibility of using openEHR archetypes for mapping and annotating extracted data from a practice based electronic health record to enable outcome research on the collected data.

1.4 Methods and material

1.4.1 Method

In order to investigate the feasibility of using openEHR for research repository, we have implemented a prototype. We used the design paradigm to construct the system, which is rooted in engineering discipline. Applying this paradigm, we iterated through a series of steps to implement and test the designed prototype systems. The applied method of iterative and incremental method of building software involved the process of specifying a problem, transforming the problem statement into a design specification, and repeatedly inventing and investigating alternative solutions until a reliable, maintainable design that fulfils the requirement was achieved.

The feasibility of the solution was assessed using a previously developed criterion. We assigned value to each variable in the criterion using evidence collected during the application development process. The value assignment is elaborated in the result and discussion chapter. However; the values assigned to the criterion depend on the experience and knowledge of the developer of the solution and this could lead to some subjectivity in the evidence used. In spite of this limitation, we were able to get valuable information about the feasibility of the approach using the developed prototype.

1.4.2 Material

Different clinical modeling and XML (Extensible Markup Language) mapping tools were used. A detail description of the materials used during the development of the archetype based research repository is given in the method chapter.

1.5 The organization of the thesis

Chapter 1: Introduction

Chapter 2: Theoretical framework

This chapter gives a general idea about the current situation regarding interoperability in electronic health records. It also discusses what the challenges are in this area and what have been proposed as a solutions to this problem. Then it briefly describe openEHR standard. It also presents related studies that have been used as a base for the thesis.

Chapter 3: method and material

This chapter outlines the method and materials used for developing the prototype and the criteria used to assess the feasibility of the developed system. It also discusses the limitations of the method used.

Chapter 4: requirement

This chapter presents the requirements. It specifies the actual functional and non-functional Requirements. It also details the approach used for collecting the requirement specification.

Chapter 5: design

This chapter describes how we designed the application, the archetypes and template needed. In this chapter, we also included the class and sequence diagram of the application.

Chapter 6: Implementation

This chapter discuss how the design was implemented using a particular technology.

Chapter 7: Result and Discussion

In this chapter, first the observations and results of the experiments will be presented. Then we will discuss the feasibility of the solution based on the results.

Chapter 8: Conclusion and Future Work

This is the last chapter and concludes the research and offers suggestions for future work.

Chapter 2

Theoretical Framework

In this chapter, we give the formal definition of the term interoperability and electronic health record, then we discuss the concept of interoperability between electronic health record and research repository. We also detail current challenges and solutions to interoperability and present solutions given by HL7, CEN and openEHR, which are implementers of standard in healthcare.

We discuss openEHR in detail since it is the open specification on which our prototype is built. This chapter includes the concept of openEHR's two level modelling approach, the specifications in openEHR, openEHR EHR structure, archetype and templates which are used for modelling healthcare knowledge. It also presents previous studies that have been used as foundation for our research. Finally we explain openEHR.NET and OceanEHR, which are .NET implementation of this specification.

2.1 Interoperability

International Organization for Standardization(ISO) is the world's largest developer of standard and it is currently a network of the national standards institutes of 163 countries (ISO 2012). ISO defines Electronic health record (EHR) as:

"*The basic-generic definition for the EHR is a repository of information regarding the health status of a subject of care, in computer processable form*" (TC 2008).

As stated in the definition, EHR contains computer processable information and this could be analysed with software to discover relation among different variables (TC 2008).

Another similar definition of EHR is:

"*Digitally stored health care information about an individual's lifetime with the purpose of supporting continuity of care, education and research, and enduring confidentiality at all times*" (Iakovidis 1998).

The definition above states that EHR should support research and education by sharing information. One of the important features of EHR is its ability to share information seamlessly with a research repository; thus EHRs need to be interoperable with external system. Interoperability between systems can be achieved at different levels (Tolk 2006). As a result, the term interoperability can have different meanings in literatures and we explicitly state the definitions used in this paper.

ISO definition of interoperability

"*The ability of two or more applications being able to communicate in an effective manner without compromising the content of the transmitted EHR*" (Begoyan 2007)

The definition used in our context emphasizes two levels of interoperability. The first one is functional or syntactic interoperability and this refers to the ability of two or more

systems to communicate or exchange information so that it is human readable at the receiver side. The second level is semantic interoperability and it is the ability of systems to share and understand information at the level of formally defined domain concepts. The information at this level is computer processable or automatically interpretable by the receiving machine.

Institute of Electrical and Electronics Engineers (IEEE) also defines Interoperability as

The ability of two or more systems or components to Exchange information and to use the information that has been exchanged (Radatz et al. 1990).

This definition also mentions the two levels of interoperability. In the definition, the idea exchange of the information is equivalent with technical/functional interoperability. The ability to use the information corresponds to semantic interoperability. The two concepts are interdependent and we need both of them to bring substantial benefits. In fact, in order to achieve semantic interoperability we need technical/functional interoperability as a prerequisite Stan (2010)

Syntactic interoperability is achieved when systems are capable of communicating and exchanging data. At this level specified data types, data structures and communication protocols are fundamental. For example, standards like XML provide syntactic interoperability.

SemanticHEALTH specified three levels of semantic interoperability (Stroetmann et al. 2009) (Stroetman et al. 2009):

- **Level 1 technical** : This refers to functional or syntactic interoperability discussed above.
- **Level 2: orthogonal Levels of partial semantic interoperability**

  - **Level 2a: unidirectional semantic interoperability**

    It is a unidirectional interoperability. The receiver can interpret and understand the information from sender's perspective.

  - **Level 2b: semantic interoperability of meaningful fragments**

    Sender and receiver can process and understand the information in meaningful parts or fragments.

- **Level 3: Full semantic interoperability**: Data from external and local system can be processed and combined identically. Seamless integration of information between different systems.

Semantic interoperability is implemented using metadata, which are used to describe data, metadata can use ontologies. Ontology is formally defined as:

A formal, explicit specification of a shared conceptualization (Gruber et al. 1993).

The ontology should be machine-processable ("formal specification") and should reflect consensual knowledge ("shared"). An explicit specification means that the concepts and relationships are given explicit names and definitions (Dogac et al. 2007).

Building a research repository involves extracting, transforming and loading (ETL) data to the target platform. This process is often the most time-consuming part of building a research repository project (Eggebraaten et al. 2007).

Extracting: The first part of the ETL process involves extracting data from the source system. This includes capturing a snapshot of a chosen subset of the source data for loading into the research repository. Data cleaning is also done during this phase including; fixing errors, misspellings, incorrect field usage, missing data, duplicate data, and inconsistencies can be resolved at this phase. Data cleaning can be done manually or automatically using technologies like pattern recognition to upgrade data quality.

Transforming: This stage applies transformation rules to the extracted data to derive it for loading into the target platform. Transforming involves mapping of the extracted data format to the research repository data format. The mapping can be done in a semi-automatic way using mapping tools that generate code.

Loading: This phase loads the data into the end target. Depending on the requirements of the system, this process varies widely. Some solutions may overwrite the existing data in a repository with a cumulative information. The frequency of updating the repository with extracted data can be done on daily, weekly or monthly basis.

While accomplishing the process of extracting, transforming and loading data from the EHR to the research repository, we need to achieve both syntactic and semantic interoperability between the systems.

2.1.1 Challenges on interoperability in EHR

Currently, EHR systems are widely used in hospitals and primary care centres (Chen 2009). However, it is usually difficult to share information and to collect data for clinical research due to distribution, heterogeneity and autonomy in healthcare information systems.

The types of data used by retrospective clinical research include; symptoms, treatment and outcomes, laboratory, and diagnosis data, which are generated from heterogeneous sources (Razavi 2007). The heterogeneity is due to varieties of platforms, database, and software applications used in healthcare. These data is mostly stored and collected from EHRs. Aggregating varieties of data in EHR from heterogeneous information resources is a common problem that a researcher has to face (Embi et al. 2009). The ability to gather records from diverse sources and to interpret the records correctly has a direct impact on the quality of the results of scientific research (Chen 2009).

Besides heterogeneity, distribution results in a severe interoperability problem in the healthcare informatics domain (Eichelberg et al. 2005). On the other hand distribution and heterogeneity is needed in research as it allows to incorporate a more diverse study populations and reduces the bias induced by any individual researcher (Iavindrasana et al. 2008).

The other challenge towards interoperability is design autonomy. EHRs can have different proprietary information models and inconsistent data quality. Likewise research repositories collecting patient data for clinical research have similar problems with different information models and terminologies (Chen 2009, Eichelberg et al. 2005). Approaches to build information model can be divided in to three categories (Bird et al. 2003).

- **Unstructured approach**: In this approach, the EHR is simply a warehouse filled with unstructured text. This approach is stable with change in the healthcare knowledge, but it does not allow to perform fine grained query so it is not that much useful.
- **Big model approach**: In this approach, the EHR is built by having a separate table or class for each clinical concept. These systems tend to have big schemas and it is difficult to maintain; as a result, the sustainability of the model is in question. The model need to be easily up-datable when clinical knowledge changes (Garde, Hullin, Chen, Schuler, Granz, Knaup & Hovenga 2007).
- **Generic approach**: In this approach the reference model has rich capabilities; it is generic enough to store any type of clinical information. To overcome the problem of lower data quality that results from using general purpose structures, standards introduced a constraint mechanism called “archetypes”. This approach is also called two level modeling.

Due to design autonomy, information model design of EHR and research repository

can be based on any of the above approaches; as a result, it is challenging to achieve interoperability between systems for a more effective clinical research.

2.1.2 Solutions

Now there is a general consensus on standards as a solution to achieve interoperability between systems (Scott 2009). Standards and open source implementation can help in achieving interoperability between EHR systems. In order to achieve semantic interoperability in EHR, we need the following four prerequisite and the first two are requirements for functional interoperability too (Schloeffel 2004)

1. Standardised EHR reference model – it is concerned with semantics of EHR information structures
2. Standardised service interface models – this deals with interfaces between the EHR service and other services including demographics, terminology, access control and security services
3. Standardised domain-specific concept models - archetypes and templates for clinical, demographic and other domain-specific concepts
4. Standardised terminologies – the language of health, which are used in the archetypes.

Standardization of data is compulsory in order to achieve semantic interoperability in a domain Maldonado et al. (2009). Matching clinical data to codes in controlled standardised terminologies is the first step towards achieving standardisation of data for safe and accurate data interoperability (Qamar & Rector 2007) (Qamar et al. 2007).

Many standards have been developed by different organizations as a solution to interoperability; thus currently achieving interoperability between these different standards is another challenge. However; technologies like semantic mapping can facilitate interop-

erability between different standards (Berges et al. 2011) (Berges et al. 2010). Semantic mediation assists in achieving interoperability by converting healthcare messages defined in one standard format into another (Martinez-Costa et al. 2010) (Berges et al. 2010).

ISO produces specification and requirements for an Electronic Health Record Reference Architecture (Begoyan 2007). The main users of this standard will be developers of EHR architecture standards such as CEN, HL7 and openEHR (Eichelberg et al. 2005)(Sanroma et al. 2004).

In the following section of the thesis we will discuss the solutions that have been given by the three developers of EHR architecture CEN, HL7 and openEHR. Since openEHR was used for building the prototype, it will be discussed in detail.

2.2 CEN

CEN is a European Committee for Standardization and it is involved in developing standards for health care systems and their interoperability. CEN covers European Union (EU) countries and some affiliated countries outside the EU (Begoyan 2007). CEN's Technical Committee produced a standards related with medical informatics called CEN 13606. CEN 13606 defines architecture for communicating the EHRs of a single patient. The standard makes sure that the meaning is correctly interpreted at the receiver side as intended by the sender of the record and the confidentiality of the data is not violated. The standard is a specification for exchange of EHR; it does not specify the internal structure or model of an EHR (Schloeffel et al. 2006). This standard uses openEHR archetype methodology (Begoyan 2007) (Leslie 2008) (Schloeffel 2004).

CEN 13606 has five parts consisting of a Reference Model, an Archetype Interchange Specification, Reference Archetypes and Term Lists, Security Features, and Exchange

Models (Schloeffel 2004)

The Reference Model: defines the generic and stable building blocks of the EHR (Begoyan 2007). Reference model deals with interoperability at the syntactic level and contains the packages stated below (Kalra 2006) (Sanroma et al. 2004).

- The Extract package: defines the EHR EXTRACT root class of the reference model and the EHR data that it contains.
- Demographics package: provides a data set to define the persons, software agents, devices and organisations that are used within the EHR EXTRACT.
- The Terminology package: includes the definitions of terms used within the EHR.
- Data type packages: defining the representation of data values for various data types including quantities, text, and basic types.
- The Access Control package: it defines a representation for EHR access policies.
- The Message package; this class is responsible for the attributes that will be required to communicate the EHR EXTRACT to a requesting process via a message.

Archetype Interchange Specification: This is concerned with archetypes which will constrain the Reference Model class. Archetypes, which are like meta-data, define patterns for the specific properties of the data. The archetypes are used to model the domain knowledge and they address issue related with semantic interoperability (Begoyan 2007) (Kalra 2006) (Sanroma et al. 2004).

Reference Archetypes and Term Lists: this specification provides vocabularies for attributes, and archetypes to represent HL7 specialised Acts and openEHR specialised ENTRYs (Sanroma et al. 2004).

Security Requirements and Distribution rules: these part cover specifications related with data protection and security when exchanging patient related medical information(Kalra

2006) (Begoyan 2007).

Exchange Models: it gives a set of models that can be used for message-based or service-based communication (Kalra 2006)(Begoyan 2007).

2.3 HL7

HL7 stands for Health Level Seven and It is one of several American National Standards Institute (ANSI) accredited Standards Developing Organizations with members of over 55 countries. HL7 is implemented in America, some European, Asian countries and Australia. The main objective of HL7 is to standardize the data exchange or communication between different types of healthcare applications. The HL7 standard supports two messaging protocols: Version2 and Version3 (HL7 2012) (HL7 2007) (Courtney 2010).

HL7 messages are generated as a result of events in healthcare; for instance, this could be an admission of a new patient. When an event occurs in an HL7 compliant system, a message is sent to the requesting application by collecting the relevant data from the applications (Begoyan 2007).

HL7 Version 2 does not support interoperability between healthcare applications very well. This is essentially due to the lack of a defined information model. In addition to that definitions for many data fields are vague and provide many optional data elements. Consequently, it was difficult to achieve interoperability between different implementation of this standard (Begoyan 2007).

HL7 Version 3 is developed to give solutions to problems in Version 2 by focusing on specific contexts, terminology, models and conceptual definitions, and relationships. The Reference Information Model (RIM) is the foundation of the HL7 Version 3 development process. It uses an object oriented development methodology (Shafarman 2012).

RIM defines the data content needed in a clinical or administrative context and gives an explicit representation of the semantic relations that exist between the information carried in HL7 messages. It is a shared model from which all domains create their messages (Gunther Schadow 2006).

Though the RIM was not intended for the purpose of database design, it provides an integrated model for health-care data, and it was found to be a suitable basis for a data model used in data warehouse architecture (Eggebraaten et al. 2007).

HL7 template is a set of structural and content constraints on HL7 static model RIM. Its purpose is to define the data content required in a specific clinical or administrative context. Templates are used to refine existing HL7 models within a narrower and more focused scope. They use terminology and ontologies to describe domain concepts in a computable way. This concept is similar to archetypes (Atalag 2007).

HL7 Clinical Document Architecture (CDA) is a document mark-up standard that defines the structure and semantics of clinical documents that are to be exchanged (Dolin et al. 2001). It is the main strategy of HL7 for EHR interoperability. CDA derives its semantic content from the shared HL7 RIM and uses the HL7 Version 3 Data Types which are defined in the RIM (Atalag 2007) (Dolin et al. 2001).

However; It was argued that the clinical content of HL7 messages is often restricted due to the fact that the messages are developed to support the administration of patient care instead of supporting the work of clinicians (Atalag 2007).

2.4 openEHR

openEHR specifications are developed by an independent nonprofit community. openEHR's objective is building an easy to create and share electronic health records by consumers using open-source and standard-based implementations. Technically, openEHR is con-

cerned with creating open source software and tools for such a platform. On the clinical side, it aims at creating high quality and sharable clinical models called archetypes along with an interface to terminology (Kalra et al. 2005) (openEHR 2007).

2.4.1 openEHR's two level modeling approach

openEHR has pioneered a two level modelling approach for EHRs (Beale 2002). Before explaining the approach, we will illuminate the concepts knowledge and information.

Knowledge: is a general statement or set of models describing our understanding of the world and the statement applies to all entities in a population (Coiera 2003) (Beale et al. 2006*a*).

Examples:

- When measuring heart rate; rate, rhythm and position of patient should be measured.
- The concept Blood pressure is composed of Systolic and Diastolic measurements in mmHg and possible positions when measurement been taken are: Lying, Sitting and Standing.

Information: is a statement about specific entities in a population on a certain circumstance and the statement may not apply for the general population (Coiera 2003) (Beale et al. 2006*a*).

Examples:

- XY's heart rate was measured to 80.
- Blood pressure measurement of patient X was 120/80 and patient X was sitting.

In single model approach both domain experts and developers are involved in specifying the requirement when developing an electronic health record system. For example to record information on the concept "blood pressure" the health professional notify the software developer what is needed to record the concept blood pressure; the systolic and the diastolic pressure, the units in which it is measured (e.g. mm[HG]), and where it is measured (arm, leg etc.). Using this information, developers design the schema for recording the concept blood pressure. In this approach, knowledge about the concept blood pressure is mixed with the schema used to store patient information. The problem with this method is that when the knowledge about the concept blood pressure get changed; the schema of the database has to be redesigned. With this approach it is difficult to cope with the dynamic nature of knowledge in healthcare; as a result, two level modelling approach was developed and in this methodology knowledge and information are separated (Beale 2002) (Beale 2003) (Gok 2008).

As shown in the Figure 2.1 we have two broad categories of models, ontologies of information and ontologies of reality which are developed and maintained by different type of authors. Ontologies of information model what we want to communicate and store in the information system about the reality. Two level modelling approach divides ontologies of information in to information model and knowledge model (Beale et al. 2006*a*).

The information model corresponds to semantics that are invariant and stable across the domain including basic data types and data structures. It is represented using openEHR reference model while the knowledge model is concerned with the dynamic domain level content descriptions of information structures such as “blood pressure” and these will be modelled using archetypes and templates in openEHR which are easy to maintain. Only the first level (the information model) is implemented in software, reducing the dependency of deployed systems and data on clinical content definitions which frequently changes with knowledge in medicine (Beale et al. 2006*a*).

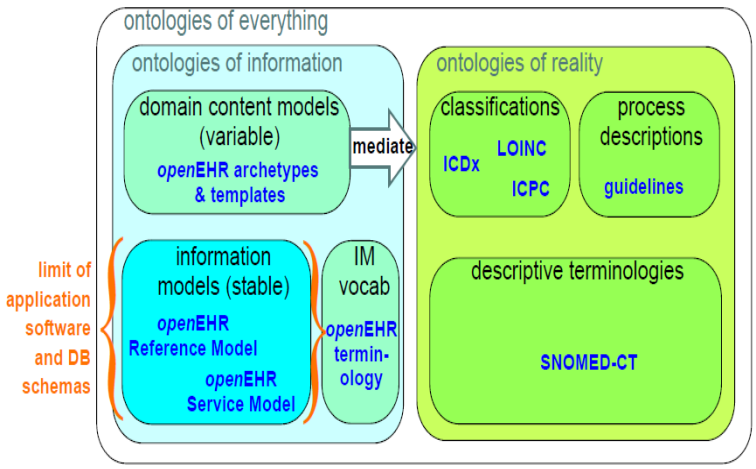


Figure 2.1: Ontological Landscape (Beale et al. 2006*a*).

The two level modelling approach also separates ontologies of realities from ontology of information to have a sustainable electronic health record system. Ontologies of reality (terminology) are used to model the real phenomena in the world. For instance terminology like SNOMED-CT(Systematized Nomenclature of Medicine–Clinical Terms) describes real phenomena by providing terms, synonyms and definitions covering diseases, findings, procedures, etc. Archetypes, which are knowledge model, are used as a gateway to easily bind with ontologies of reality. Separating these three models helps to limit the dependence of one on the other, resulting in more maintainable and adaptable systems (Beale et al. 2006*a*).

2.4.2 OpenEHR Package Structure

Reference Model (RM), Archetype Model (AM) and Service Model (SM) are the three major packages that are defined in openEHR specification (Beale et al. 2006*a*).

2.4.2.1 Reference Model (RM)

The actual instances of information are created from the reference model. It defines and provides identifiers, data types, data structures and various common design patterns. It enables access to knowledge resources and provides support for archetyping. It also defines the semantics of EHR and demographics (Beale et al. 2006*a*).

Fine-grained structures or "bottom level" models defined in the Support and Data types are used in the Data Structures and Common models. Data structures and common models are used in turn in the "top level" models like the EHR, EHR Extract, Demographic and Composition (Beale et al. 2006*a*).

2.4.2.2 Archetype Model (AM)

The archetype Model enables description and creation of archetypes and templates which represent clinical knowledge to be captured in an openEHR based system. Under Archetype model specification, the package Archetype and template define the object-oriented semantics of archetypes and templates respectively (Beale 2008).

2.4.2.3 Service Model (SM)

Services an openEHR EHR is expected to offer to the users are defined in this model. It includes the package(Beale & Heard 2003):

- Virtual EHR API(Application Programming Interface): defines the fine-grained interface to EHR data at the level of Compositions and below.
- EHR Service Model: defines the coarse-grained interface to electronic health record service and level of granularity is openEHR Contributions and Compositions.
- Archetype Service Model: defines the interface to online repositories of archetypes.

- Terminology Interface Model: provides the means for all other services to access any terminology.

Figure 2.2 shows the relation between the parts of the computing platform and the three specifications described above.

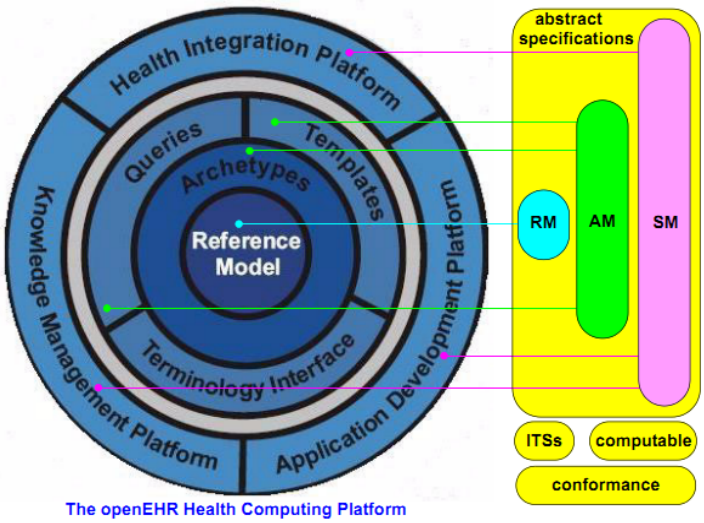


Figure 2.2: openEHR specification (Beale et al. 2006*a*).

2.4.3 OpenEHR EHR structure

openEHR's EHR is structured as a simple object model and corresponds to the record of a single patient; it includes the objects; EHR, EHR Access, EHR Status, Directory, Contributions and Compositions structured as shown in the Figure 2.3 (Beale et al. 2006*a*) (Beale et al. 2006*b*).

- EHR: this represents the root object and it will be identified using a globally unique EHR identifier
- EHR Access (versioned): this is an object containing access control settings for an EHR

- EHR Status (versioned): an object holding status and control information. It optionally includes the identifier of the patient associated with the record.
- Directory (versioned): these are hierarchal folder and their purpose is to logically organize Compositions. For instance dividing compositions into compartments related to a fixed episode of care.
- Compositions (versioned): these correspond to the containers of all clinical and administrative content of the record.
- Contributions (versioned): It is the change-set records for change made to the health record.

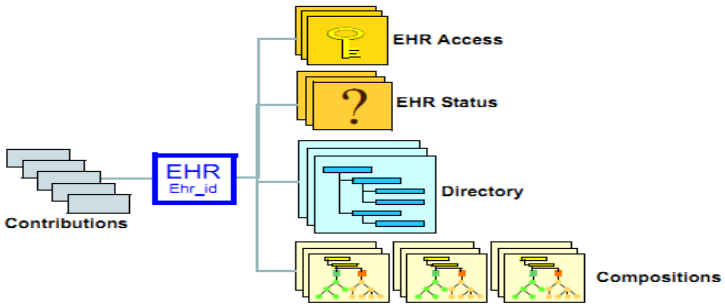


Figure 2.3: High level structure of the openEHR EHR (Beale et al. 2006*a*).

In openEHR, created information is eventually expressed using "Entries". An Entry is logically a single 'clinical statement' and it can have a significant amount of data. There are five subtypes under the class ENTRY and these are ADMIN ENTRY, OBSERVATION, EVALUATION, INSTRUCTION and ACTION. Among these classes, the last four are classified under the same category CARE ENTRY.

Compositions can have different ENTRIES including; OBSERVATIONS, EVALUATIONS, ACTIONS and INSTRUCTIONS organized using SECTIONS. Under these ENTRIES, we can have data in different data structure such as LIST or TREE of ELEMENTS and CLUSTERS which in turn can have different data types. Figure 2.4 shows a typical

composition logical structure hierarchically from composition to data types.

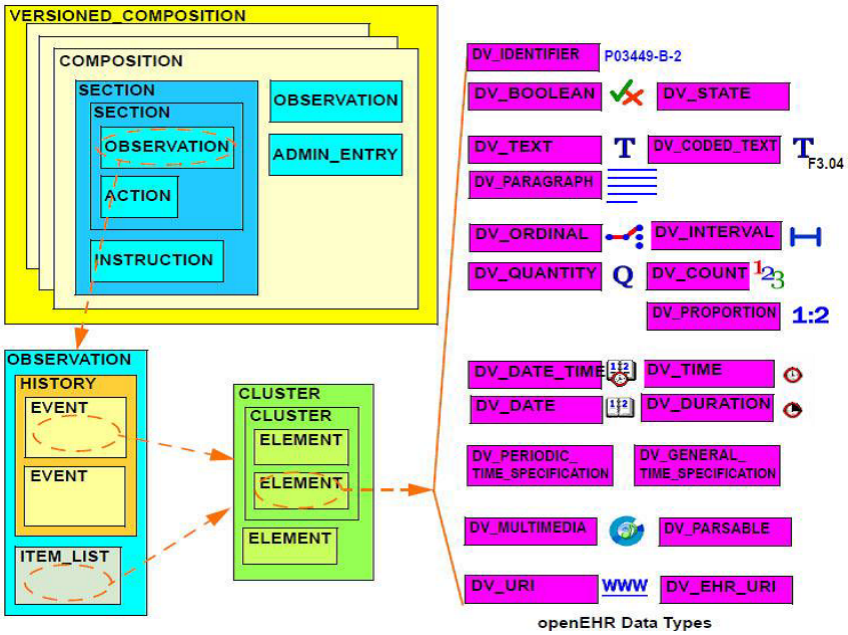


Figure 2.4: Elements of an openEHR Composition (Beale et al. 2006*a*).

2.4.4 Archetypes and Template

The most important concept introduced by openEHR is the "archetype" concept (Beale 2002). It is a machine readable description of how to store data using the Reference Model. An archetype, in openEHR, can be considered as a model for recording clinical information. Archetypes are stored separately from the data in their own repository. The creation and modification of the content and querying of information conforming to the openEHR Reference Model (RM) is controllable by archetypes (Leslie & Heard 2006).

The concept of archetype was explained in a good way by Thomas Beale using the "semantics" of Lego bricks. The set containing all the possible combination of bricks is vast; however, most of these combinations are meaningless or only small portion of this set holds the interesting object like a car and house. Equivalently, the Reference Model

defines a vast informational construction space, only small proportion of which contains combinations valid in the domain. The valid Lego brick constructions can be the result of imaginations or printed plans included in Lego packages. Such plans are similar with the concept archetypes (Beale 2002).

As shown in Figure 2.5, In the archetype approach an EHR system needs to offer three building blocks (Beale 2002).

- **Archetype Editor**: it is application for creating and editing archetypes.
- **Validator**: a component, which creates or manipulates valid data using archetypes.
- **Browser**: this corresponds to generic data browser or editor that can be built, and this can be built based on the reference model or the archetype model class.

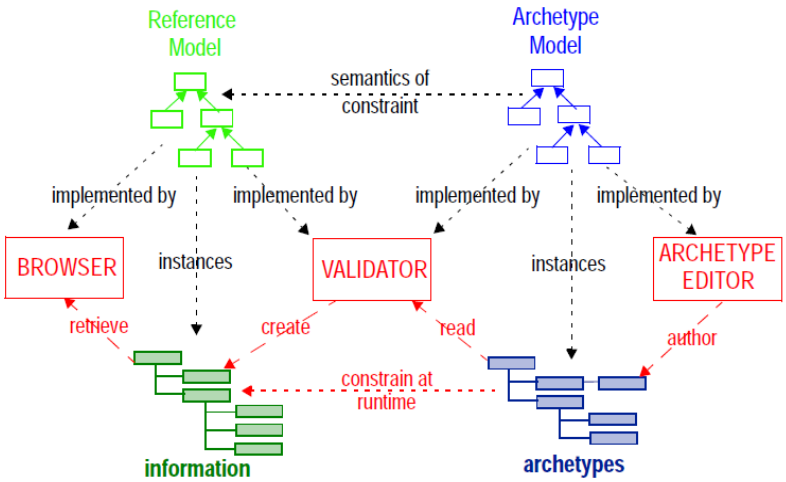


Figure 2.5: openEHR archetype methodology (Beale 2002).

Archetypes can be divided in to two broad categories

- '**Legacy**' **archetypes**: These archetypes mimic legacy data and they do not follow any ontological design. Mostly they are flat, or else tree-like and are designed by technical persons.

- '**Designed**' **archetypes**: Clinicians are not passive users of openEHR-enabled software, but they actively participate and determine the possible breadth, depth, and richness of data in EHR systems through the role they play by creating and revising designed archetypes. Designed archetypes include the archetypes in the openEHR Clinical Knowledge Manager (CKM).

Heather Leslie and Sam Heard presented methodological approach for developing archetypes and it includes the steps (Heard & Leslie 2008):

- Identifying clinical concepts
- Identifying existing archetypes that can be re-used wherever possible or modify if necessary
- Creating new archetypes if necessary

The major computational function of archetypes is to support querying. The paths taken out from archetypes are the basis for queries into the data. Queries in openEHR are specified in AQL, which is basically based on Structured Query Language (SQL) and XPath style paths extracted from archetypes. The AQL has two major innovations (Ma et al. 2007):

1. Using the openEHR path mechanism to represent the query criteria and returned results.
2. Using a 'containment' mechanism to state the data hierarchy and constrain the source data to be queried.

The following is an example AQL query meaning "get the doctor description for diagnosis code of C18.2 for a specific patient":

```
{SELECT c/content/description/items[at0014]/value/value
FROM EHR e[ehr_id/value=$ehrId]
CONTAINS Composition c[openEHR-EHR-COMPOSITION.testdischarge.v1]
CONTAINS Evaluation diagnosis [openEHR-EHR-EVALUATION.testcolon.v1]
WHERE diagnosis/data/items[at0002]/value/defining_code/code_string ='C182'}
```

Archetype Object Model (AOM)

Archetypes are instances of Archetype Object Model. AOM defines a language in which to write archetypes. When an archetype is represented in memory, the archetype will exist as instances of the AOM classes. In serialised form, archetypes can be represented in various ways. The most common serialisation in openEHR is Archetype Definition Language (ADL) (Beale 2008).

An archetype definition includes three parts: descriptive data, constraint rules, and ontological definitions. The descriptive data contains a unique identifier for the archetype, information about the archetype such as author, version, and purpose. The constraint rules specify restrictions on the valid structure, cardinality, and content of the EHR component instances complying to the archetype. The ontological part defines the controlled vocabulary (i.e., machine readable codes) that can be used in specific nodes in instances of the archetype (Eichelberg et al. 2005).

Figure 2.6 shows the relationship between ADL and AOM. As shown in the figure the archetype parser reads ADL and represents it as instance of the AOM class.

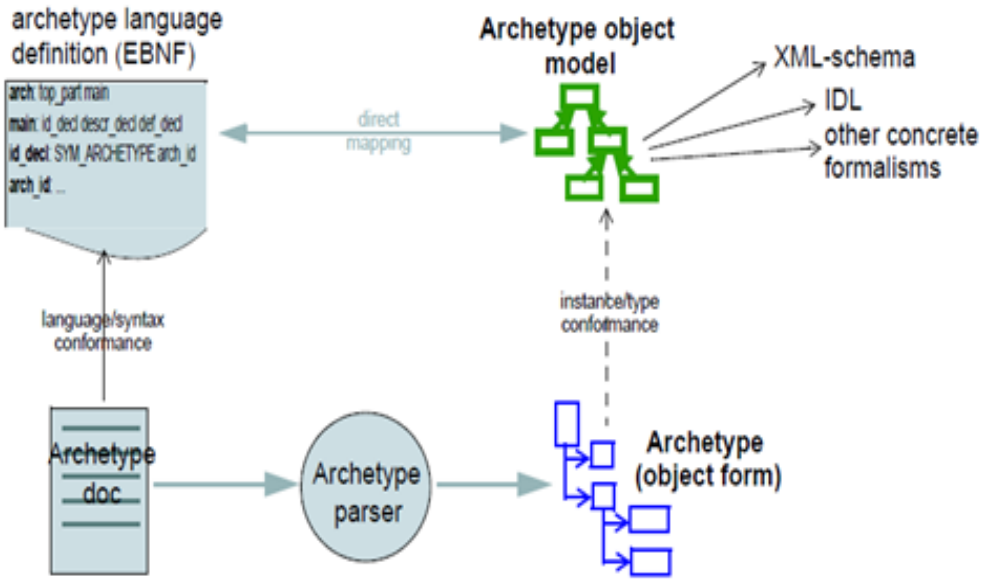


Figure 2.6: relation between ADL and AOM (Beale 2008).

Template

Templates specify a tree of one or more archetypes, each constraining instances of various reference model types (Leslie 2008). For instance there are archetypes for concepts like " diagnosis " (an Observation archetype), templates are used to aggregate more than one archetypes together to form Compositions in the EHR, e.g. for "discharge Report" which can contain observation archetypes like diagnosis organized using section archetypes. Templates can correspond to data to be imported to openEHR platform (Heard et al. 2003).

Templates are basically developed and used locally, while archetypes are usually widely used. A template is used at runtime to validate data input, guaranteeing the data is valid with respect to the constraints defined in the archetypes referenced by the template. In particular, template conforms to the path structure of the archetypes referenced, as well as their terminological constraints (Beale & Heard 2005).

2.4.5 openEHR.NET

openEHR.NET: enables to build object using Reference Model (RM) and archetype model (AM) classes. These classes allow developing software that can produce and processes clinical information that is compatible with the clinical models developed by domain experts. This information can be serialized to XML and send to a server such as OceanEHR. Currently, there are three projects; openEHR, EhrGate.WSClient and EhrGate.WsClient.Tests under this software (serefarikan 2012) .

openEHR: This project includes the RM and AM class implementations, along with the utilities and it is used build RM objects(serefarikan 2012) .

EhrGate.WSClient: This project has proxy classes which allow sending data in RM classes to OceanEHR web services. EhrGateWsClient take data from the created Composition instance using openEHR project and put it into Composition proxy generated from WSDL (Web Services Description Language) and send it to web service (serefarikan 2012) .

EhrGate.WsClient.Tests: this project contains unit tests that provide overview of the functionality that the OceanEHR web services API provides (serefarikan 2012) .

OceanEHR: is a platform for deploying Electronic Health Records and applications. It has interface which enable application to have access to different services including EHRs, demographics, terminology and security services. OceanEHR can import information from structured and unstructured data sources. EHRs are stored in EhrBank and can be queried with archetype-enabled query language. The AQL processor can query a single EHR or the entire EHR population in the server for research studies and it can return whole Compositions or fine-grained data items (informatics 2009) .

2.5 Related work

A literature review was done to identify how to design and construct openEHR research repository using currently available tools. In this part of the chapter we will detail previous works that have been used as a base for our work. There was a study which compared a proprietary template model called Julius system with similar approach including archetyped based system and they recommend that future system for research and quality management should consider using the openEHR models with shareable archetypes (Chen & Enberg 2007) besides another study had shown that proprietary EHR models can be represented with archetypes without losing much semantics. They suggested that importing legacy archetypes in to the proprietary templates can facilitate secondary use of data (Chen et al. 2009) (Chen 2009). These two studies were the starting point of the thesis.

We also found an interesting study which had an overall aim of examining how to introduce openEHR architecture based EHR system to an emergency department in a hospital (Gok 2008). Figure 2.7 below shows how an EHR system based on the openEHR approach could be integrated with existing system. The application has a component called converter and it provides interfaces for the systems and parses the received messages to the correct format for the EHRbank. In this study, the data was transferred via template data schema (TDS) based XML files.

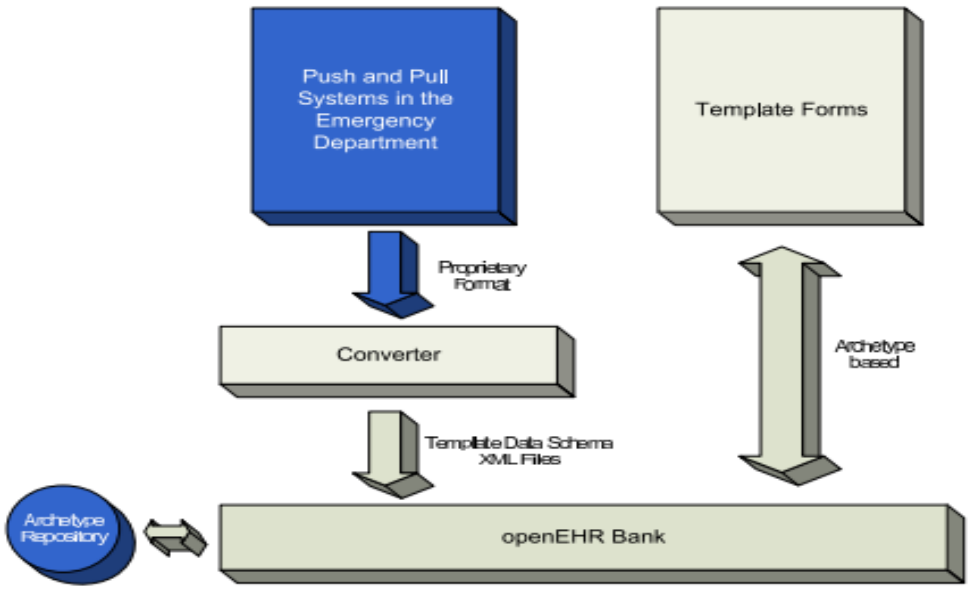


Figure 2.7: an openEHR-based EHR system (Gok 2008) .

There was also a related work we found that had an overall aim of publishing existing medical data stored in a relational database as prEN 13606 compliant EHR extracts that conform to existing Archetypes (Rinner et al. 2007). The task was divided in to two tasks; first they published relational data as EHR Extract using direct XML publishing approach. Then XSLT (EXtensible Stylesheet Language) approach was selected to transform the resulting EHR Extract into an archetyped EHR Extract. The XSLT script was generated using visual mapping tool.

The other important study we found was with the title towards use of OpenEHR Archetypes to support views of Cystic Fibrosis Review Records (Corrigan 2010). The study provides a good guideline on how one can design archetypes and templates for openEHR application.

2.6 Summary

In the first part of this chapter after defining the concept interoperability and EHR, we detailed the ETL process that we need to accomplish while building the research repository. We also discussed the current challenges to interoperability between electronic health record and research repository. After that we outlined the solutions that have been given to this problem and discussed precisely the two standards HL7 and CEN among the three main standardizing bodies currently affecting EHR implementations.

In the second part of the chapter we have discussed about openEHR general architecture how openEHR supports the two level modeling then we discussed the three specifications in openEHR and we also have detailed the reference information model which is used for recording the stable part of patient health record. Then we explained archetypes and templates that will be used for modeling domain concepts. Finally In this chapter we presented studies that have been used as a base for work.

Chapter 3

Method and material

The aim of the thesis is to assess the feasibility of using openEHR as a research repository. In this chapter, first we will present the method and materials used to develop the prototype, then we will detail the criterion used to assess the feasibility of the built solution.

3.1 Method

There are three major paradigms that can be used for building solution in the discipline of computing (Comer et al. 1989). But we have chosen the design paradigm as it is more suited to this problem. The method is rooted in engineering and it consists of four steps followed in the construction of a system to solve a given problem:

- State requirements
- State specifications
- Design and implement the system
- Test the system

We iterated through stating requirement, stating specifications, designing and coding steps when tests revealed that the latest version of the system did not satisfactorily meet the our expectation. We used incremental iterative method to build the system. Incremental development slices the system functionality into increments. In each increment, a slice of functionality was delivered, from the requirements to the deployment.

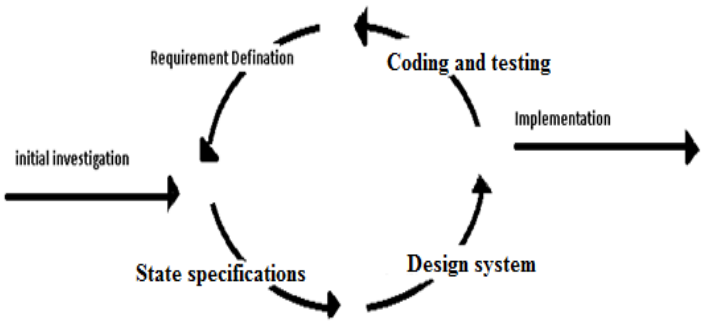


Figure 3.1: method used to develop the application.

The basic idea behind the method used to build the solution is to develop a system through repeated cycles and increment. We started with an implementation of a subset of the system requirements and iteratively enhanced and added new functionality until the full system is implemented. We have divided the project in to two slices. The first feature provided was extracting and committing data to openEHR format. Then in the next version of the application, we added annotation and visualization of the committed data to our application. In each slice, as shown in Figure 3.1, we iterated through stating specification, system design, and coding and testing. When our expectation was not met, we redefine the requirements of the system again and restart the iteration process as shown in the Figure 3.1.

As stated above, the focus of this thesis is assessing the feasibility of the developed prototype. In order to formalize the study concerning feasibility, we have used previously developed feasibility study criterion as a base (Gall et al. 2011).

The method use effort and effect to assess the overall feasibility, within each category sub criteria is applied, following an analytical hierarchical approach. The overall feasibility is then calculated as the average of the sub criteria. The sub criteria are given equal weight. Each criterion has only three distinct values ("high" , "medium" and "low"). The process of building the prototype was used as a source of data to assign a value to the criteria used in our feasibility study. The sub criteria include:

Effect

- Expected value gain: this criterion should assess the potential of openEHR research repository. How much customers will benefit once the system has been developed successfully. How well the proposed system satisfies system requirements identified in the requirements analysis phase, the capacity of the proposed system to meet the performance requirement.

Effort

- Development time frame: this assesses the time frame in which the system can be developed? This criterion evaluates how fast the system can be put in place and this was based on the time needed to build the prototype.
- Scope: This criterion deals with whether the prototype can be developed by applying only internal resources, only external resources, or both. Internal resources includes developer time and knowledge, tools and hardware used to build the solution. It is well known fact that managing external resource is more difficult than managing internal resource.
- Resource: Which resources are required to develop the system and can these resources be accessed? Do we currently possess the necessary technology? This question was answered by assessing the availability of the required resource to build the prototype.

- Complexity: this assesses whether there are multiple parties involved in building the prototype. Whether tools are available which ease the process of building openEHR applications and how easy it is to use the tools.

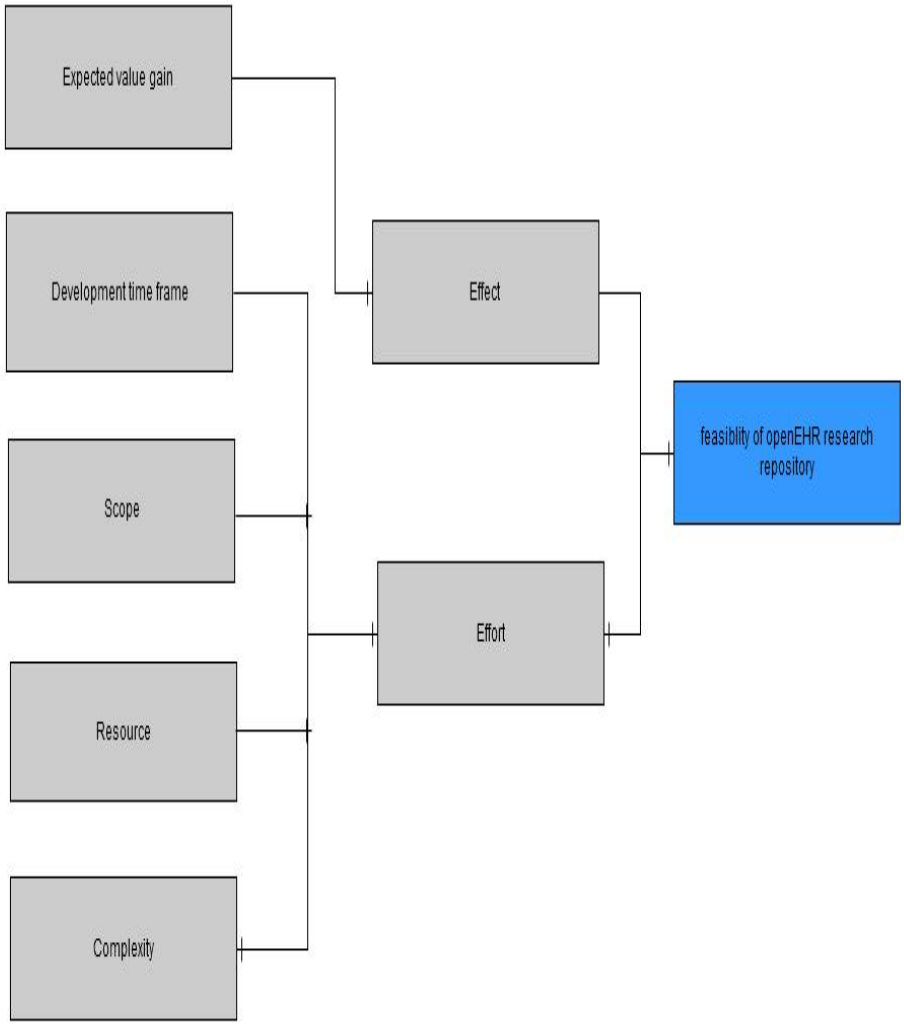


Figure 3.2: feasibility study criteria used.

3.2 Materials

This section list the materials that were used for building the prototype.

OceanEHR instance server

It is a health record computing framework which is based on openEHR release 1.01. It is designed to accept information from unstructured and structured data sources. It was used to save the data that has been extracted from DIPS database. The annotated data was also stored on this platform. Querying capabilities are available, using the Xpath-compatible AQL.

Microsoft VISUAL C# 2008 EXPRESS

We used VISUAL C# 2008 EXPRESS as a development tool and NET 3.0 as the .NET runtime. C# was used as a programming language to develop the application.

Eclipse

Eclipse was used for exporting the generated Java source code from tools as an executable Jar file.

XML mapper

It is used to generate the source code to transformXML/CSV(Comma-Separated Values) to XML file format. We used tool from Altova© MapforceMapForce as it provides a simple graphical interface to define and execute mappings based on Schema.

Clinical modeling tools from ocean informatics

Archetype Editor

The Archetype Editor is used by clinicians to design archetypes for use in clinical and research settings. It supports all of the openEHR archetype semantics. It enables clinicians and other domain specialists to model information semantics without knowing about openEHR or other technical details. Bindings to terminology codes can also be defined using the tool. Using the Editor, Archetypes can be created for any openEHR information concept including; Composition, Section structures, and various types of En-

try: Observation, Evaluation, Action, Instruction, and AdminEntry. The final archetype can be viewed in the openEHR ADL,XML, or in various formats including HyperText Markup Language (HTML).

Template designer

The Template Designer enables users to compose a set of archetypes into a collection called a template. It has a drag and drop Graphical User Interface (GUI) interface, which can be used for creating template as needed for local use. This tool can also be used to generate the operational template and this can be used for data validation.

Workbench

The work bench was used to validate the archetypes before uploading. It also enables to have a look at the path to each ELEMENT in our archetypes.

3.3 Critique of the method

The method used to evaluate the feasibility of the application was based on the assessment of the resource and effort required during the development of the application. The frame of reference was our knowledge and experiences, which may significantly differ from other potential developers. Consequently, the assessment tends to give biased results; this is because we gave our own opinion on how we think the prototype development process was achieved. In addition, measuring some of the criteria in our feasibility study need more time. In spite of these limitations, we were able to get useful information about the feasibility using the developed prototype.

Chapter 4

Requirement specification

In this section of the paper, we will discuss the requirements of the system. We will also detail the method used to collect the requirement specifications. Then we will state the assumptions and constrain under which the system operates.

Requirements are commonly classified as functional and non-functional requirement. Functional requirement is concerned with what the system do. It specifies an action that a system must be able to perform without considering physical constraints; it is a requirement that states input/output behaviour of a system (Parviainen et al. 2004).

On the other hand, non-functional requirement is mostly concerned with how the software does the functional requirements. It specifies system properties such as performance, interoperability, maintainability, extensibility, and reliability. In this chapter, we will discuss both the functional and non-functional requirement of the system (Parviainen et al. 2004).

4.1 Assumption and constrain

The aim of this thesis as stated in chapter one is to assess the feasibility of openEHR as a research repository. OceanEHR is a secure, high-performance health record computing framework. It is implemented using Microsoft.Net and SQL Server 2005 as the underlining technology. Though OceanEHR is a specific vendor implementation of openEHR specification, we assumed that it can provide general feedback about the feasibility of openEHR standard implemented research repository.

An important requirement while building healthcare research repository is the security and privacy aspects of patient information. We assumed that privacy, consent, and security issues are already addressed. These can be resolved by removing information related with patient identity; however, in this study we used false data that would not put a patient's privacy at risk.

Extracting Data is a complex process which involves obtaining a snapshot of source data that is appropriate for research purpose in the knowledge domain. We assumed that this task had already been done manually by executing the appropriate SQL queries and the data is available as CSV file. Extraction is assumed to be done in a long time interval and it was also anticipated that the extracted data will over write the data in the research repository.

While developing the system there were constrain on implementation technology. Since the platform is implemented in .NET, we were forced to use C# as our programming language and VISUAL C# 2008 EXPRESS as our development tool.

4.2 Source of requirement

It is well known that there are a number of techniques for identifying requirements. We had three sources of requirement while collecting the requirement specification for the system.

The first source used was brainstorming and the identified requirements had been discussed with the potential users of the system. A set of requirements usually is best gathered from the interaction of a group where ideas are shared and developed. In addition, we have done literature review to observe what requirements need to be fulfilled while building a research repository for healthcare.

The other important source of requirement was users. Users are the key element of the requirement collection phase. They are principal system drivers providing their intentions, desires, or problems to the requirement specification process. We have presented what we planned to develop to one researcher on gastrointestinal surgery and one telemedicine and ehealth student. Then we asked them for a possible feedback and functionality they want to have on the developed system. It helped us to make the interaction between the developers and users effective, and work together to jointly define the requirements. Since the number of users used as a source of requirement is limited, we have also used personas to collect requirements representing a group of users.

4.3 Methods used for requirement collection

Methods used for requirement collection depends on a number of factors, such as availability and location of stakeholders, customers' and users' knowledge of the problem domain, and development team knowledge of the problem domain (Scott 2004).

The methods used for collecting the requirements are meeting with users and personas.

Since developers knowledge in the problem domain is low, it was better to use the techniques mentioned above to collect the requirements of the system.

We frequently talk about the users of the system in general while we design software, but that is too general. It is better to discuss with real users of the product and how they would react to our product. To collect detailed requirements, we used personas. Personas define an archetypical user of our system, an example of the person who would interact with it. For this purpose, we develop two personas Alen and Fred. The personas represent descriptions of typical users of our systems along with stories about how they would use the product to meet their goals. Then we asked ourselves “How do we help Fred, the surgeon and Alen, the telemedicine and ehealth student to get what they want more quickly?"

Figure 4.1: Fred.

Fifty-one year-old Fred is the father of two children and the grandfather of one. He lives with his wife. Once a month he meets with his children. He is a medical doctor and has specialized in surgery. Currently, he is doing research in healthcare.

He wonders if he will be able to find out what is going on in the hospital more easily. He is also interested in reading journals on healthcare which are mainly written in his field of specialization. His focus is mainly on surgery that is performed on colon. He spends his time looking for outcome on surgery related information. Fred is mostly interested in observing correlation among demographic data, discharge date, admission date, diagnosis found and treatment given.

He wants to learn computer tools, but not at the expense of his time for his research work.

Fred wonders if he would be able to cope with the computer or information system in healthcare. He does not mind asking his children for help when he wants to fix problems on his computer, but asking the guys at work for help is another story.

Figure 4.2: Alen.

Alen is a master degree student in telemedicine and ehealth and has experience of working as a medical doctor. Alen is 28 years old, single and lives in his private house.

He sends a weekly report to his supervisor; once a month he meets with his supervisor to discuss the status of his research work. Alen likes to use computers as a tool and to install new software in his Macbook Pro and accomplish tasks using them. He also wants to discover all the functionality that the software provides.

He spends most of his time reading journals which are related with healthcare. He is also highly interested with the concept of archetypes and motivated to develop and contribute archetypes. Besides he is interested in observing relation between diagnoses found and treatments given to a patient.

From those two personas we can collect requirements of the application. We can infer that Fred want to be independent in his work environment; as a result, it is a requirement making the system less effort demanding to understand and use. In addition to that Alen may be interested in affecting the semantics of the extracted data through design archetypes. The persons also give input to AQL queries design.

We have used the method shown in Figure 4.3 below from IEEE to verify system requirements collected at this phase of the development cycle.

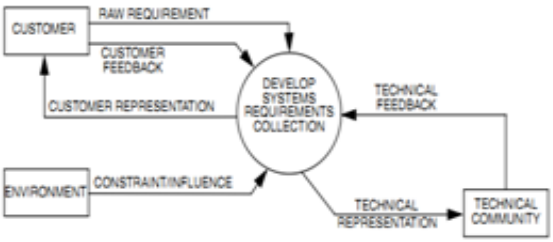


Figure 4.3: requirment varification method (IEEE 1998).

Besides verifying the requirement with the above technique, the entire application verification and validation was done using a method from Food and Drug Administration (FDA). FDA stresses the need for software verification and validation before putting software into use. As stated in validation guidelines all software product should have a requirement specification that defines its intended use. Verification should be carried out at the end of each phase of software development cycle. At requirements capture phase review of document with user was used to check for the accuracy of the requirements if they are properly interpreted by the developer (FDA 1997). Figure 4.4 shows method from FDA used to verify and validate the application.

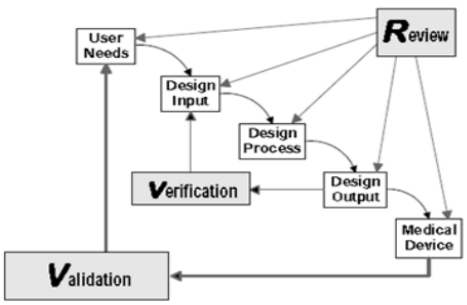


Figure 4.4: application varification method (FDA 1997).

In general software verification and validation are difficult because a developer cannot test forever, and it is not easy to know how much evidence is enough. Software validation is about developing a “level of confidence” that the software fulfils all requirements and

user expectations for the system automated functions and features. We have used manual testing in order to fully test that the requirements of the software are met. In this method of testing, software tester acts like end user and works on all the options provided by the application to ensure correct behaviour.

4.4 Functional requirement

In this section of the paper, we will discuss the functional requirement of the system. We have made an event list and used a use case diagram to illustrate these requirements.

Event list: In order to systematically enumerate the functional requirements, we made use of an even list. The event list contains all the events that can take place in the system and the possible inputs/outputs at each event. The requirement in the event listed in table 4.1 are also illustrated using the Unified Modeling Language diagram of the use cases in Figure 4.5. The use case shows the interaction of the system with the users of the implemented prototype.

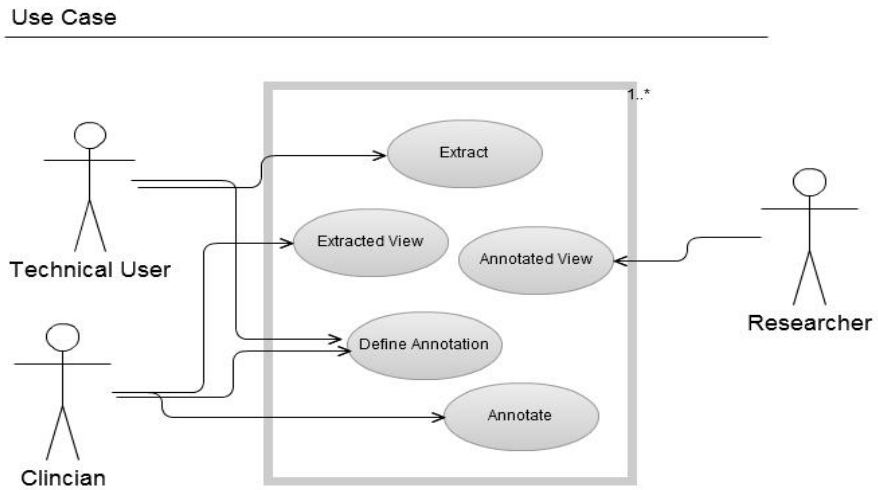


Figure 4.5: use case

Table 4.1: Event List

Event name	Input	Output	Description of task
Extract data	CSV file data	OpenEHR format data based on generic archetype	This functionality allows importing data from the EHR to the research repository. Technical person can extract the data.
Define automation	• XSD based on generic archtype • XSD based on design archetypes	• Mapping Rule • Design archetypes • Template	The task includes using Archetype Editor by Person with domain knowledge like Alen to generate design archetype. Technical person will generate mapping code to transform the data.
Annotate Data	• openEHR format data • Mapping Rule	OpenEHR data based on design archetypes	Enable users like Alen to annotate the data with the design archetypes (metadata)
Visualize data	Query	Visualization	Visualize the data in the openEHR platform. Enables researchers like Fred to have view of data.

Below for each use case is given: actors involved, brief description and a priority measure from low to high which describe the importance of the use case in meeting the goal of the application.

4.4.0.1 Use case 1 Extract data

The actors in this use case can be a technical person, clinicians or researchers, but mostly technical person will extract data. This task is given high priority because it is the base for other requirements of the prototype too. This capability enables users to update the research repository with recent data from the DIPS EHR.

- The system need to transform data from the extraction format to openEHR format.

- Save the data in the openEHR platform.

4.4.0.2 Use case 2 View extracted

This use case enables users to have a view of the data imported to openEHR. The actors in this use case are mostly clinicians who will develop the design archetypes after having a look at the imported data. This task was given medium priority as this functionality can be dropped out without affecting the overall system much.

4.4.0.3 Use case 3 Define annotation

The actors in this use case are clinicians and technical person. This is a high priority task. The clinicians need to create metadata in the form of archetypes. These archetypes will be used to add important information to the extracted data that can be useful to researchers. Below I will state the action that has to be done to accomplish this task.

- The clinician should be able to add metadata in the form of archetype needed for annotating the data.
- The technical person should be able to use the metadata or the archetypes to generate the mapping rule.

4.4.0.4 Use case 4 Annotate

The actors in this use case are mostly the clinicians like Alen or Fred. It enables them to annotate the extracted data using the concepts made by clinicians.

4.4.0.5 Use case 5 View annotated

The actors of this use case are the researchers like Fred. This functionality was given high priority as it is one of the important functionality that end users need. It enables users to view the data in the openEHR platform based on their semantic queries. Fred and Alen can have different visualization and semantic query.

- Design semantic query based on their data presentation need
- The system will show the queried data.

4.5 Non-functional requirement

In addition to the functional requirement stated above, the system is expected to fulfil non-functional requirements. Below I will discuss the non-functional requirement that is needed from the system.

4.5.0.6 Interoperability

One of the most important non-functional requirements needed is interoperability between the EHR and research repository. We need to map data in the EHR to the repository without losing the meaning of the data. The quality of the data in the research repository depends on the level of interoperability between the repository and the electronic health record.

4.5.1 The general look of the prototype

The user interfaces and the general look of the application for end users can impact use of the solution. The simplicity of the look of the application is one of the important

non-functional requirements. If the system is not simple to understand, it can lose its usability. In general if end-users feel the system is not easy to use, or too cumbersome no matter how much excellent, the system could fail to achieve general usage. On the other hand an excellent interface design can make a product easy to understand and use, which results in greater user acceptance.

4.5.2 Maintainability requirement

The application need to be divided into components so that it would be easy to modify and replace components. The system components can change regularly and this change should not affect the overall system. The change in healthcare knowledge could affect the annotation and the view of the data in the repository; this should not make operating and maintaining the system a challenging task to carry out.

4.6 Summary

The first part of the chapter mentions the assumptions under which the prototype work and the constrain while developing the prototype. Later, the chapter articulates the functional requirements using a detailed event list and a use case diagram. In this chapter, we also discussed the method used for collecting the requirement. The chapter concludes with discussing non-functional requirements that the prototype is supposed to meet, and why.

Chapter 5

Design

In this chapter, we will describe the overall application design, the software architecture, and the classes that make up the application. Then we will detail the sequence diagram corresponding to the features specified in the requirement specification; finally we will discuss the design procedure used for archetypes and templates needed for the application.

5.1 Design goals and considerations

5.1.1 Modularity

We aimed at using modular programming. We added an interchangeable component called module by breaking down the application functions into modules which accomplishes one function and contains everything needed to accomplish this. It improves maintainability by enforcing logical boundaries between components.

5.1.2 Reusing existing Archetypes

One of the openEHR goals in the clinical space is about creating high-quality, reusable clinical models of content and process - known as archetypes - along with formal interfaces to terminology (openEHR 2007). To build semantically interoperable systems, it is not sufficient to just create an instance of the Reference Model; also the Archetype concept has to be shared between systems (Garde, Knaup, Hovenga & Heard 2007). In order to fulfil the above statement, we have considered using archetypes from Clinical knowledge Manger.

5.1.3 Effective use of openEHR and XML tools

Different commercial and open source tools have been developed for openEHR applications. These tools make implementation of openEHR applications easy by reducing the application development task to the usual computer science problems like XML programming. While designing the system, we aimed at using these resources effectively in the implementation phase of the project so as to ease the application development.

5.2 The overall architecture of the system

Our prototype was designed as a client server application which is one of the most common software architecture used. Our reason for choosing this architecture was the availability of ocean's remote server which is an openEHR implementation in .NET that could be used for testing openEHR applications.

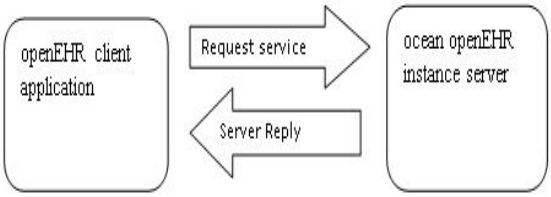


Figure 5.1: application architecture.

The client was designed as a fat client. openEHR.NET, which was explained in the theoretical framework chapter, is a middleware component and it was deployed on the client-side. The client part of the architecture was designed to handle typical user interaction, RM object building and communication with the server through web services. Data storage and management was done on the server side. Figure 5.2 shows the detailed architecture of the application which was based on the openEHR Data Conversion Architecture (Beale 2005).

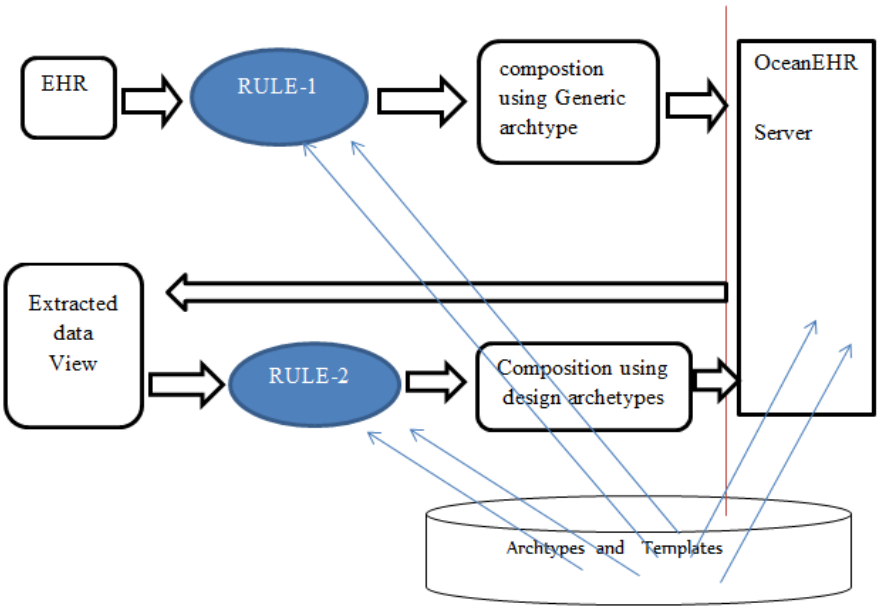


Figure 5.2: detail application architecture.

As shown in the architecture, the first important component of the application is RULE-1

which was implemented as a separate module. The purpose of this module is to map the extracted data using a flat archetype which does not need clinical background knowledge to develop. At this stage, the data is changed to openEHR format syntactically.

The second main component of the application is RULE-2 and its purpose is to map the queried openEHR formatted data to a new composition instance. This new composition is described using design archetypes created by healthcare workers. At this stage, the semantics of the data are defined.

The other alternative design was to map the CSV data directly to a template based on design archetypes. However, we have chosen the first approach since it adds more flexibility to the application by separating the tasks of syntactic and semantic mapping into two steps.

Figure 5.3 shows the functional diagram of the application. As shown in Figure 5.3, the system first puts the extracted data in openEHR platform. Then queries this data from the openEHR platform, annotates it further using design archetypes and saves it on the openEHR platform again.

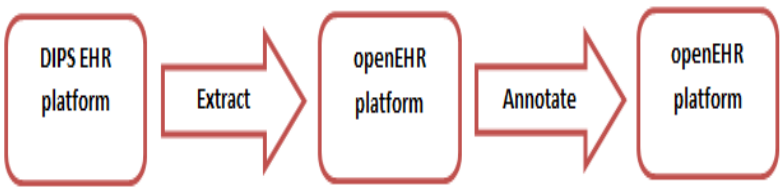


Figure 5.3: functional diagram .

5.3 Class diagram and sequence diagram

In the following part of the chapter, we are going to discuss the classes comprising the application which is illustrated in Figure 5.4. In the class diagram the lines represent the relationships. Since the structure of the application is more complicated than shown in

the figure, only important classes and methods which are necessary to understand how the application works will be discussed.

Program: This is the entry point of the program which creates an instance of ViewHome class and calls show method when the application starts.

Parser: This class provides methods for reading and writing CSV file. It has a method for reading the whole extracted file and another method for writing a CSV file corresponding to one composition.

RULE-1: It is a module which has a function of transforming the data from the original format to openEHR format. This module will take a CSV file corresponding to a composition from the CSV parser and map it to a composition. The compositions are validated using a template referencing a flat archetype which mimics the structure of the data in legacy system. Separating this module is good programming and design practice, as it allows us to modify the mapping without worrying about the other parts of the application and this will increase the maintainability of the system.

FlatCompositionBuilder: This class is responsible for running RULE-1 and generates composition in XML form.

RULE-2: This component will map the compositions made using RULE-1 to other compositions which will be validated using a template referencing design archetypes. These archetypes can be used as a gateway to annotate the extracted data. RULE-2 will be implemented as a separate project and a class called MappingMapToComposition in this project provides a method to execute the mapping.

ViewHome: This class provides buttons for interacting with users.

- *Import* button: It will start the process of committing the extracted data by transforming it to openEHR data format using flat archetype.

- *View Imported* button: The purpose of this button is to view the data that is committed to the openEHR platform using the Import button.
- *Annotate* button: starts the process of mapping the data, which is committed to openEHR using the flat archetypes, to a new composition instance. Then it will commit it again to the server.
- *View* button: Its purpose is to view the annotated data.

CompositionManger: This class uses web service methods in EhrServiceProxy to commit composition, create EHR, query EHR and initiate a connection with remote openEHR server instance.

QueryBuilder: builds AQL query from the user input.

DataStore: DataStore is used for holding the result of query from the server and the result is used to populate the GUI of the application.

QueryHandler: Sends the AQL query to the server and save the query result in the DataStore.

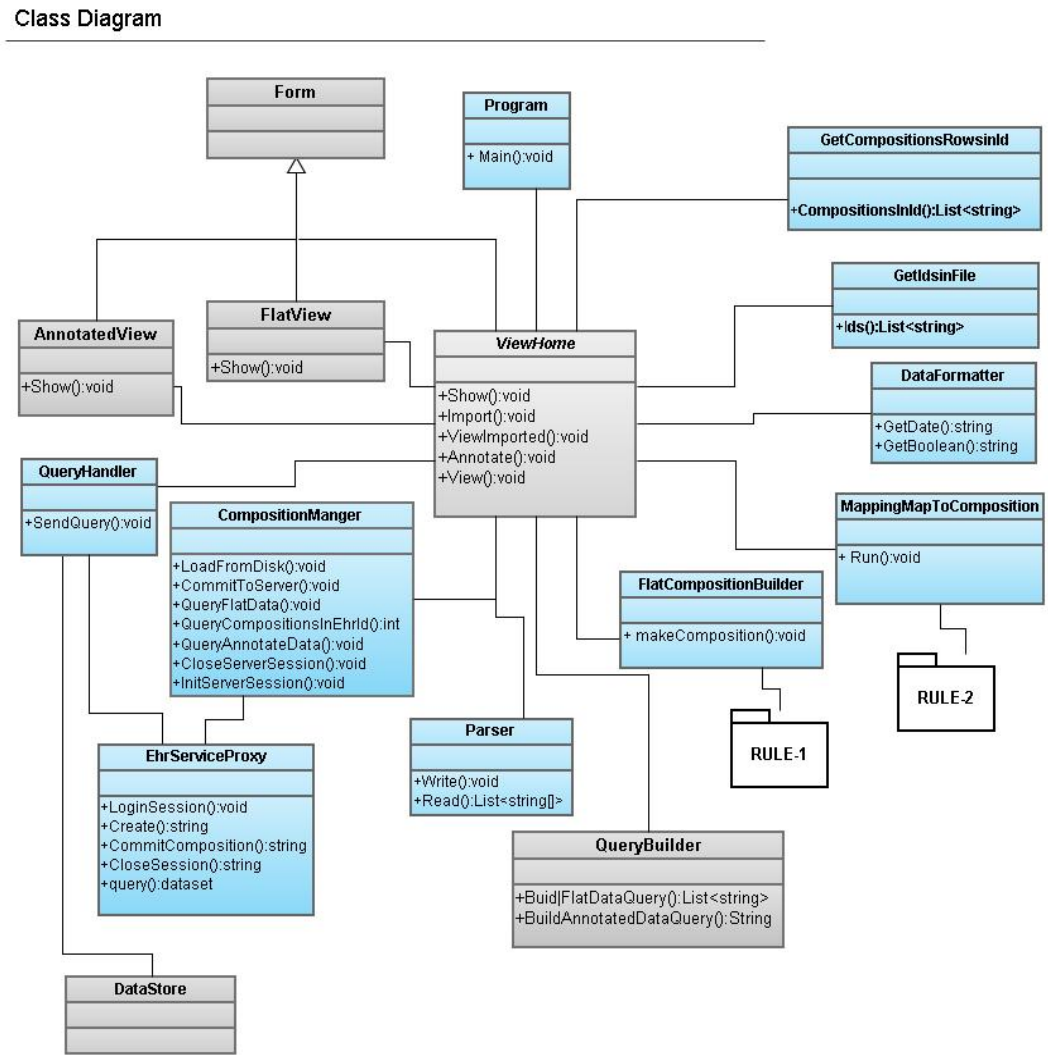


Figure 5.4: class diagram .

DataFormatter: It has methods to change the syntax of extracted data fields to openEHR format. For instance, date and boolean format are changed using methods in this class.

GetCompositionsRowsinId: This class has a method which returns all the compositions in a single patient ID. The inputs for the method are the parsed file from the Parser and a patient Id.

GetIdsinFile: It has a method which returns the entire patient ID taking the parsed file from the Parser as an input.

FlatView and AnnotatedView: These classes inherit from Windows Forms and they are responsible for visualizing the query result from the server.

In the following part of the chapter we will describe each use case and the sequence diagram associated with them.

5.3.0.1 Use case Extract

This use case involved mapping the data in DIPS EHR to openEHR data type format and we achieved syntactic interoperability at this stage. Before mapping the extracted data to openEHR format, we studied the database schema of our electronic health record and we designed mapping RULE-1 which mapped the CSV file to composition as shown in Figure 5.5. The server used a flat template referencing generic archetype for validating the composition made using the RULE-1 before committing.

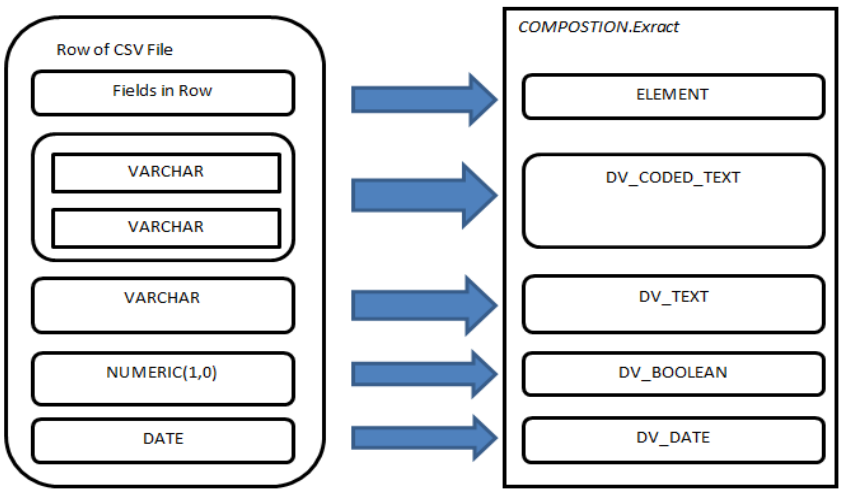


Figure 5.5: mapping between CSV and Composition.

Figure 5.6 shows the sequence diagram of the use case Extract. In this use case a user clicks on the button Import. The parser read the CSV file extracted from DIPS server. For each row in the parsed file, the DataFormater will format the fields to openEHR format. Then the parser will write the formatted fields as a CSV file. RULE-1 will map the CSV file to composition XML and we will deserialize the XML to object and commit it to the server.

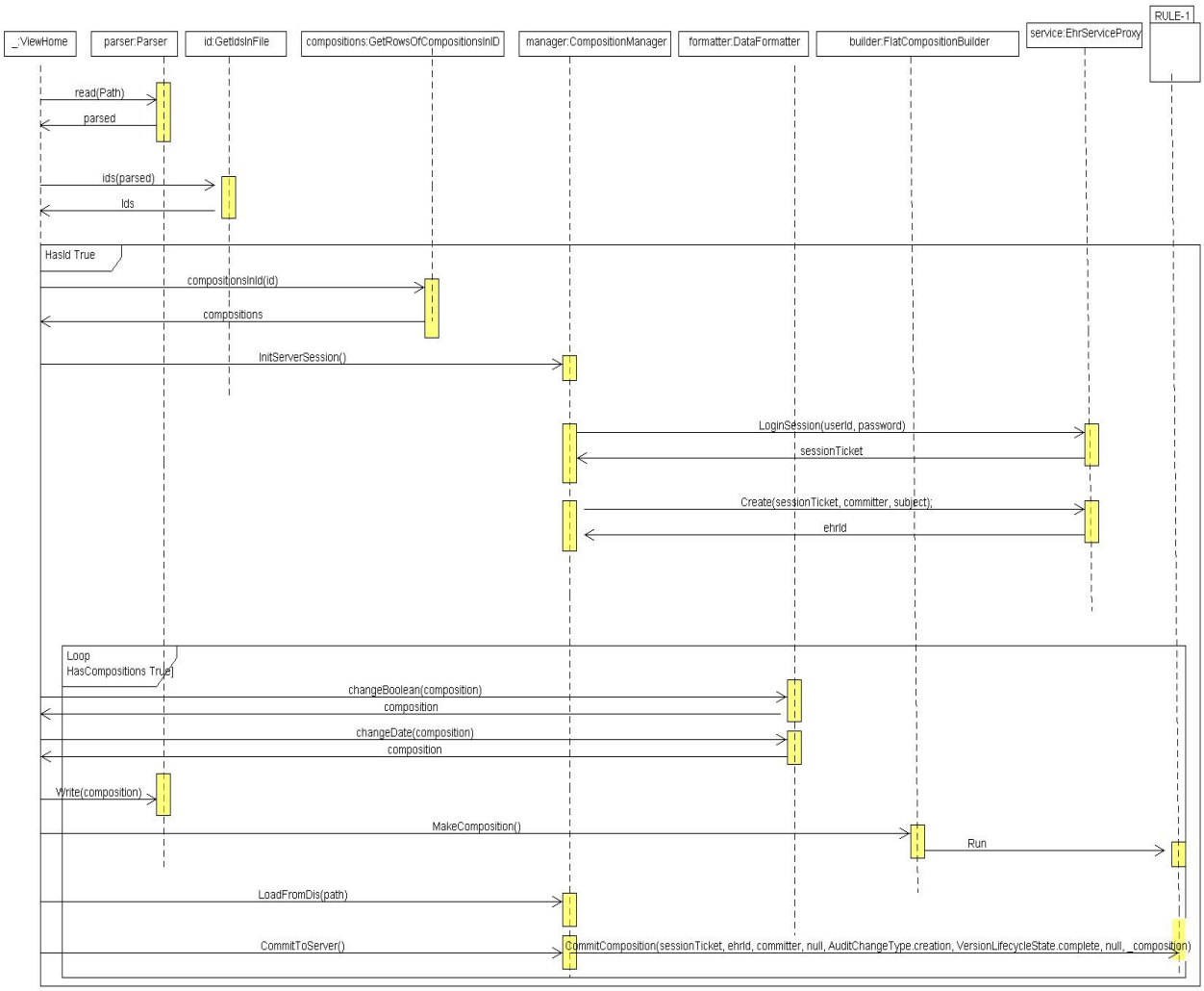


Figure 5.6: sequence diagram of use case extract.

5.3.0.2 Use case Define annotation and Use case Annotate

These use cases enabled annotating the data with semantic concept defined by the health-care workers. The first task under this use cases was defining the rule for transforming the data then integrating and using the defined rule in the application.

Defining the annotation and the rule

This comprised defining the view and transformation rule for affecting the structure and semantics of the data committed by the use case Extract.

Having a discussion with the users, we were able to get information about the data they were interested in doing the research on. Then we designed archetypes corresponding to this data. These archetypes are like metadata which can be used for annotating the data by binding with external terminology. We used conceptual mapping rule shown in Figure 5.7 to map the data with the archetypes.

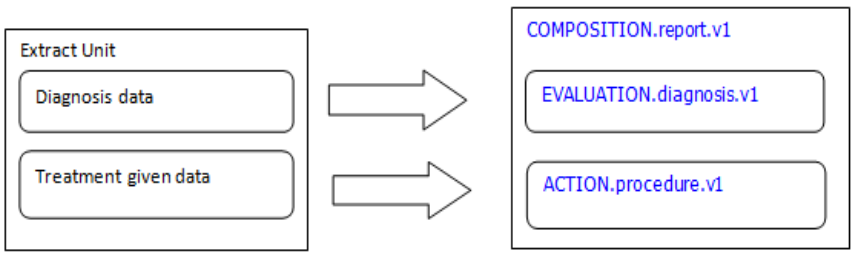


Figure 5.7: mapping to design archetypes.

This use case involved designing a template. From the template one can generate TDS and that can be used as the view of the target platform. Since we could not find the web service method to commit data using the TDS, we made composition objects using the composition XML Schema Definition (XSD) based on the template, as a view. Then we mapped the schema of the query result to this composition XSD and generated RULE-2. The server employed the template to validate the RM instance before committing.

The section, archetype and template design in this chapter, describes in detail how we designed the archetypes and templates.

Annotate

After generating the mapping rule, we integrated it with our application and used it to make the new RM instance objects that will be described using design archetypes. Figure 5.8 shows the sequence diagram of the use case annotate. In this use case, the user click Annotate after importing the data to openEHR using Import button. After that the system start to query all compositions for each EHR id committed by the button Import. Then RULE-2 will map each query result which is a list of compositions in XML to multiple compositions XML files. Each composition XML will be deserialized to object and saved on the server using the method CommitToServer. If the user clicks on annotate before importing, the user will be informed to import data first.

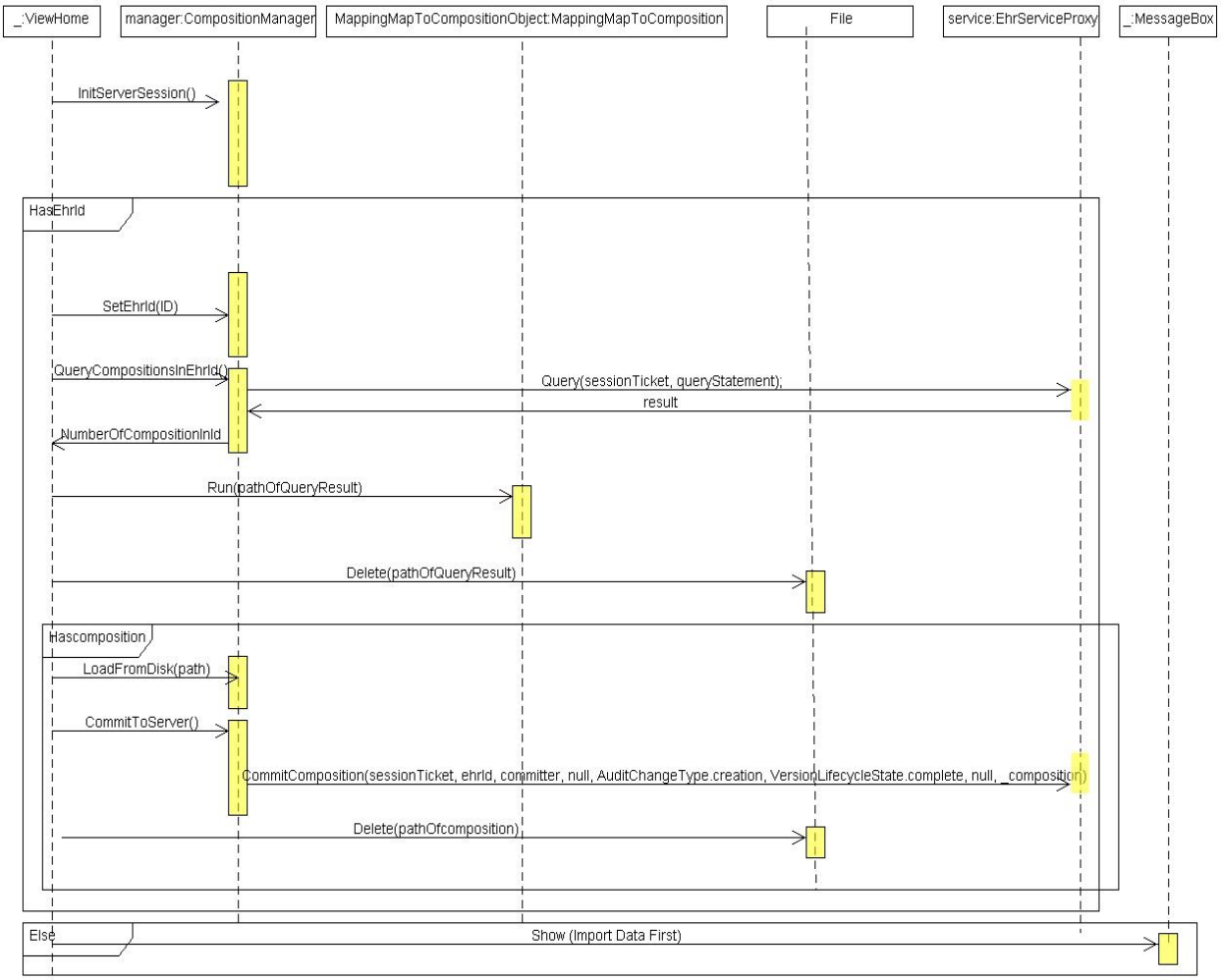


Figure 5.8: sequence diagram of use case annotate.

5.3.0.3 Use case Views

There were two use case views in our application; the first one was for visualizing the committed data by the use case Extract and the second one was for visualizing the data committed by the use case Annotate. Due to the similarity, we showed the sequence diagram of the use case that enabled users to view the annotated data using the button View.

Figure 5.9 shows the sequence diagram of this use case. In this use case, the user select query criterion and clicks on the View button. The query builder will make the AQL query using the input from the user; then the CompositionManger asks the server for an EHR id with the specified criterion and peak randomly one EHR id from the result. The compositionManger will add the EHRid with the query criterion and send it to QueryHandler. The QueryHandler will create instance of EhrServiceProxy and send the queries in parallel. and store the result in DataStore. Finally the result will be visualized using the windows form which will be initialized with value from DataStore. If the server does not return an EHR id for the criterion, the application will inform the user that there is no data for the criterion.

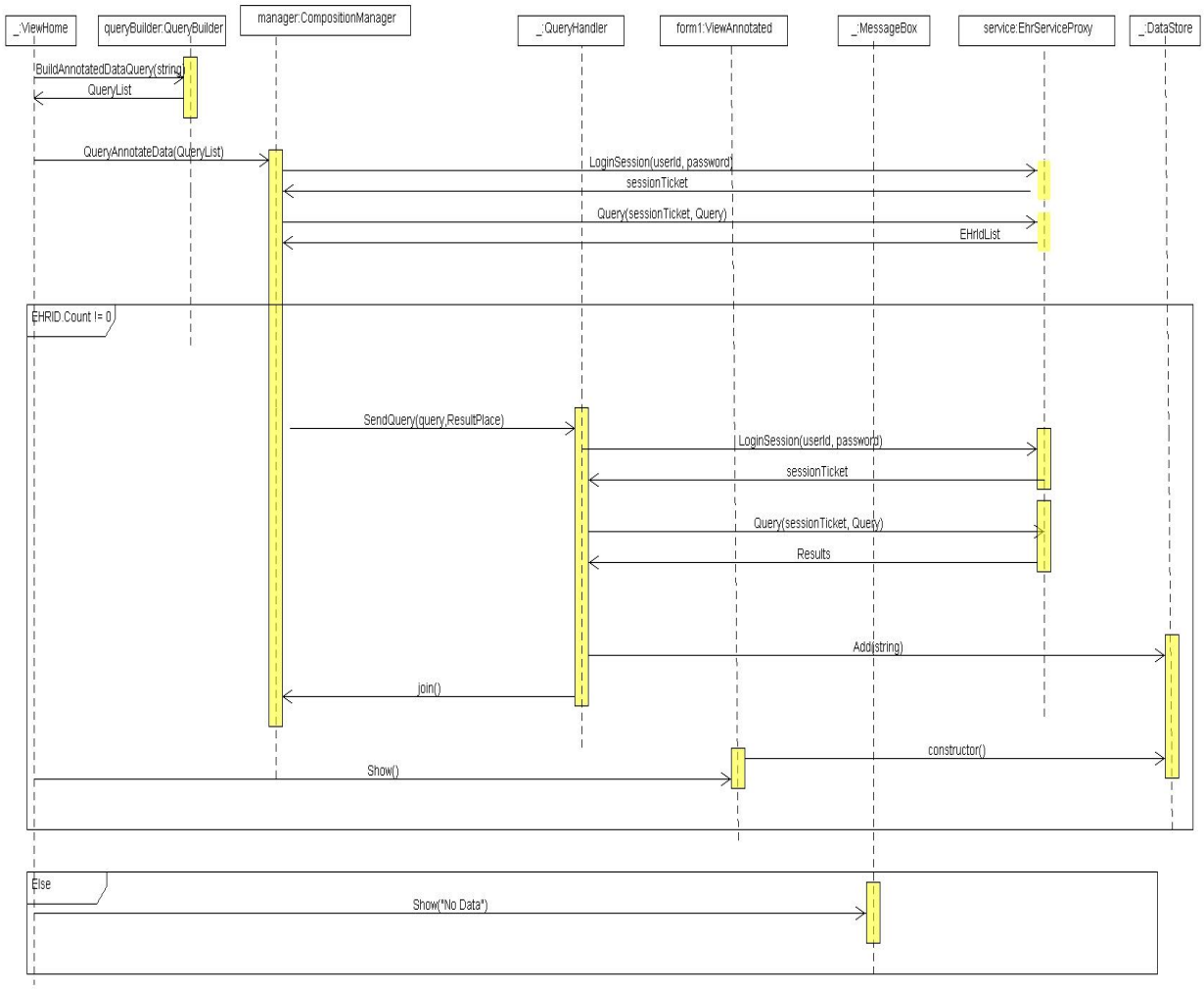


Figure 5.9: sequence diagram of use case view.

5.4 Archetypes and Template Design

We used a guideline developed to design the archetypes and templates needed for our application (Corrigan 2010) (Heard & Leslie 2008) . In the following part of this chapter, we will clarify the steps followed to design the archetypes.

1. Documents to be modeled

All data collected is entered into a database that is accessible by the clinician via the Intranet. A convenient starting point for modeling the archetypes was the schema detailing all database tables and coded values to be stored in the database. This document provides the definition of the data to be extracted and is helpful for matching to the content of existing archetypes.

2. Determine all clinical items in the domain to be modeled

After having a discussion with the users, we were able to select the data they were interested on. They were interested in information related to diagnosis and treatment given on the diagnosis malignant neoplasm of colon (ICD-10 code of C18). Table 5.1 states the content of data extracted.

Table 5.1: Data to be modelled

Clinical concept	Description of data
Diagnosis	ICD code of diagnosis found in the patient, doctor's description of the diagnosis and other related data
Treatment / operation	It is data about treatment procedure associated with each diagnosis. It includes doctor's description, date of operation, result whether the operation was successful, and other related data.

3. Map the derived clinical concepts to existing archetypes

A key concept stressed in the guidance documents issued by openEHR is the objective of re-using existing archetype designs (Heard & Leslie 2008). openEHR CKM, which is an internet based repository of existing freely available archetypes, is used as a source of existing archetypes. We were able to find archetypes from CKM that can be used to describe diagnosis and treatment given.

Action archetype was used for modeling the treatment given and its purpose was to capture services given or activities performed. Figure 5.10 shows MindMap diagram of

the archetype.

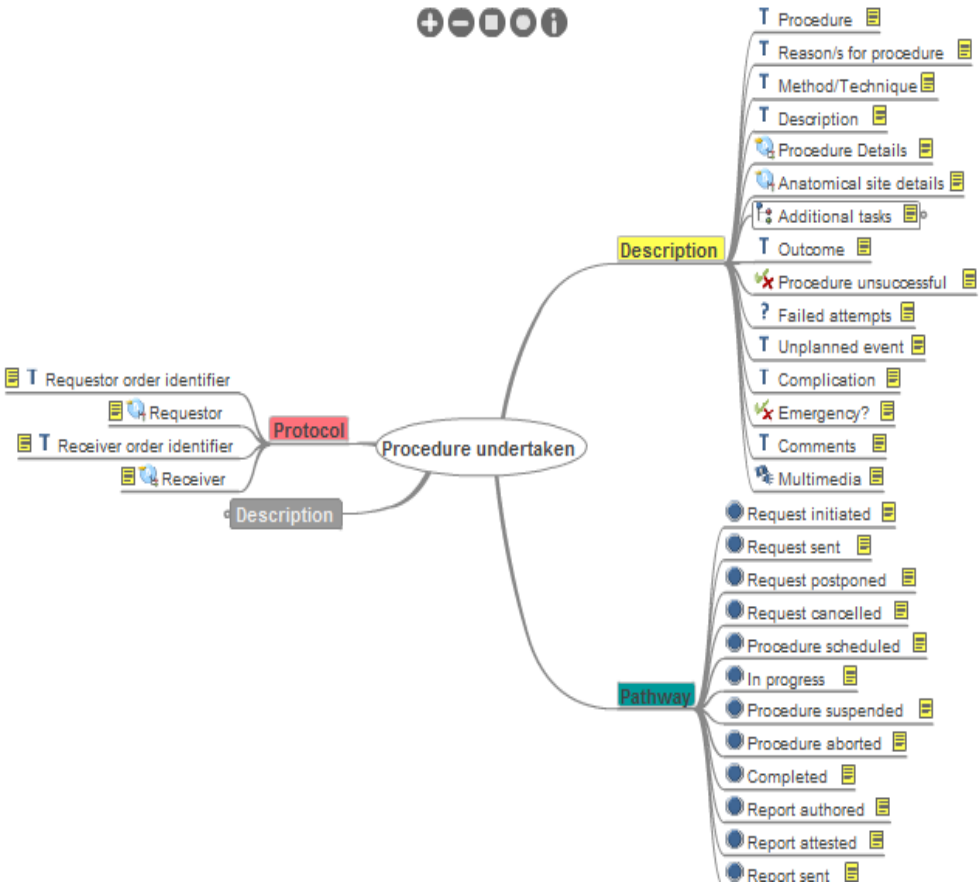


Figure 5.10: Mindmap of Action archetype used.

The archetype used from CKM for describing diagnosis was of type EVALUATION. It was used for recording diagnoses that were generally based on Observation evidence. This archetype constrained the Evaluation class. Figure 5.11 shows MindMap diagram of the archetype.

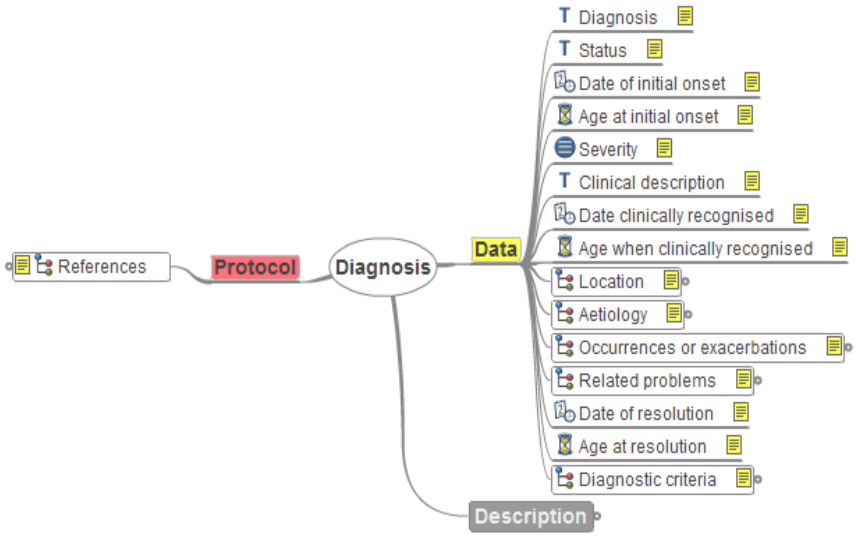


Figure 5.11: Mindmap of Evaluation archetype used.

We designed two composition archetypes and the first one provides a slot for the flat archetype. The second composition archetype was used for structuring the ACTION and EVALUATION archetypes. As stated in chapter two, a composition is the unit of information that can be saved against an electronic health record for a patient.

Template:

After creating the required archetypes, the next step was designing the templates. We build the templates using the compositions as a container. The basis of our generic template was the EXTRACT composition archetype which provided a slot to our flat archetype. The second template used the composition archetype Report and it provided unrestricted slots to the ACTION and EVALUATION archetypes discussed above.

A separate development tool called the Ocean Informatics Template Designer was used to design the templates. Ocean Informatics Template Designer offers different views of templates. The server validated the data using templates before committing.

5.5 Summary

In this chapter, we have first discussed the design goals. After that we detailed the client server architecture of the application and reason for choosing the architecture.

We also discussed the class diagram of the application and showed the sequence diagram of the use cases. Then we clarified how the data was mapped from our EHR to the openEHR research repository. Finally, we explained the procedure used for selecting and developing the archetypes and templates used in the application.

Chapter 6

Implementation

The design was implemented as Microsoft .NET windows application. We used VISUAL C# 2008 EXPRESS as a development tool and NET 3.0 as the .NET runtime. At the time, the application was developed these were the latest technologies that openEHR.NET supported. Furthermore the module, RULE-1 ,was implemented as a Jar file and Java Runtime Environment 6 was used as a running environment.

OpenEHR.NET is a software that enables to implement openEHR application that produces and processes clinical information that is compatible with clinical models. we downloaded the software from CodePlex and built it on the client machine.

Under this software, OpenEHR and EhrGate.WsClient are the important projects used while implementing the application. OpenEHR enabled us to build object using Reference Model (RM) classes. EhrGate.WsClient allowed us to send the contents of RM objects to OceanEHR server using web service methods.

As these projects were written in C#, we were constrained to write the main application using C# as our language. The web service methods used to interact with the server include: service for creating EHR, querying, committing composition, closing session,

and login session.

Below shows an example code which uses the login session web service method:

```
service = new EhrServiceProxy(url);
sessionTicket = service.LoginSession(userId, password);
```

To use the web service method, which commits composition, composition objects have to be made from the extracted data. As discussed in the design section each row in our extracted file contains information about a diagnosis and treatment given for that diagnosis, and it was logically mapped to composition.

There were two possibilities to make compositions. The first option was to write C# codes manually to make the composition objects. The second approach was to create the XML file which conforms to the composition XSD and deserialize the XML to RM object instances using the project OpenEHR.

We used the second approach since XML tools can be used to simplify the application development. As a result, the project involved mapping CSV and XML file to XML file. To accomplish this, we have looked at different solutions. We were able to get different visual XML mapping tools which provide an automatic generation of source code for files based on their schema. With this generated code it is possible to produce the XML files. Table 6.1 summarize our search result for the tools. The functionality the tools provide is marked with "X". Further analysis of tools for Schema mapping can be found in (Robertson et al. 2005).

Table 6.1: Mapping Tools

	CSV toXML		*XML* toXML	
Tool	Java	.NET	Java	.NET
Altova MapForce	X	X	X	X
Stylus Studio Convert toXML	X	X	X	
Liquid Data Mapper			X	X

In this project, we used tool from Altova© Mapforce since we found the tool and the

documentation easier to use. It is a commercial tool that must be purchased, but we used the trial version for the purposes of this study. By visually associating the elements in the input file Schema with the corresponding element in the output file Schema the mapping tool creates Java or C# source code.

Figure 6.1 shows how the application was implemented. As shown in figure, the application has two main components or modules called RULE-1 and RULE-2.

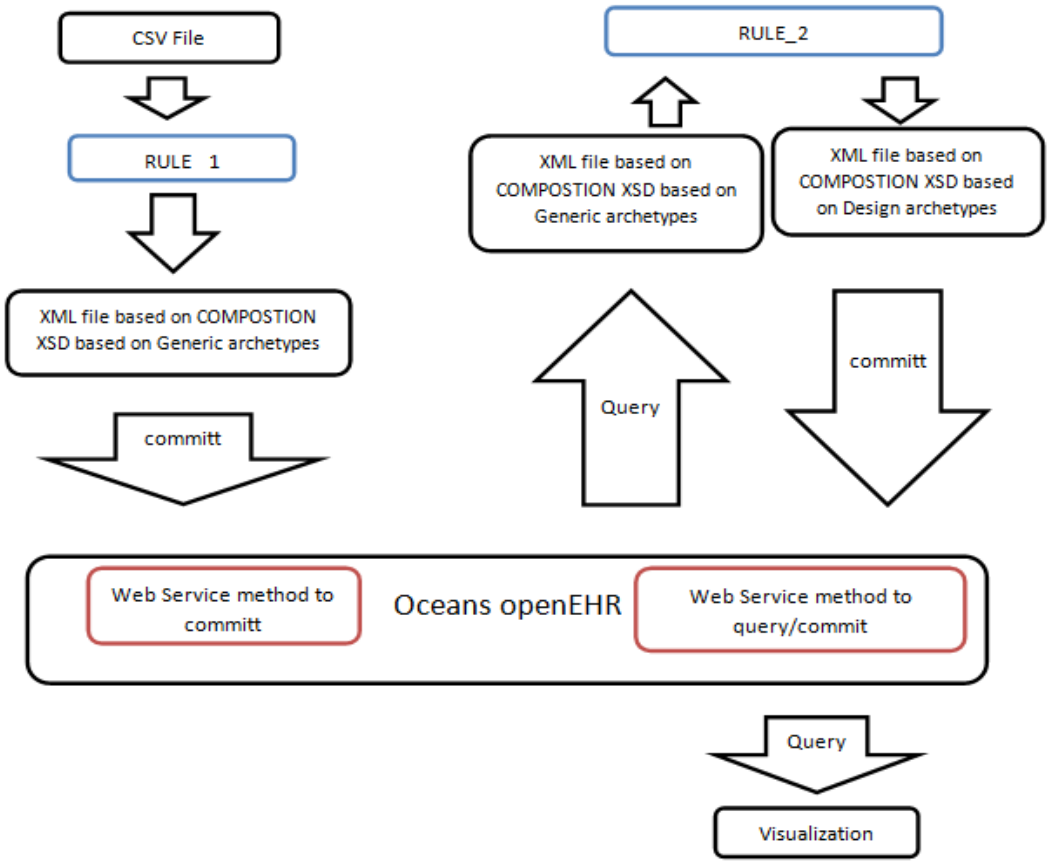


Figure 6.1: overall application implementation.

RULE-1 is responsible for transforming the extracted CSV file into the XML file that conform to the composition XSD based on the flat template. RULE-1 was generated using the visual mapping process as shown in Figures 6.2, on the figure the CSV schema

is shown on the left side and the composition XSD is shown on the right side. The finished mapping was used to generate the source code. The tool was needed for mapping; the transformation of the data was done using the generated source code. By executing the resulting code, the values defined during the mapping process were retrieved and resulted in a XML document that was compliant with the target XML Schema, containing the data selected from the CSV file using the first Schema.

We set the input and output file name for the generated Java source code and we exported this source code as an executable Jar file using Eclipse. We used this approach because it was easy to change the Jar file when we needed to modify the mapping.

We executed the Jar file using the console with the C# application and the produced XML file was deserialized to RM object using the OpenEHR project. Then the composition objects were sent via the EhrGate.WsClient project and it was validated at the server side using the flat template that referenced the legacy archetype. We have attached an example XML composition produced using this module at appendix A.

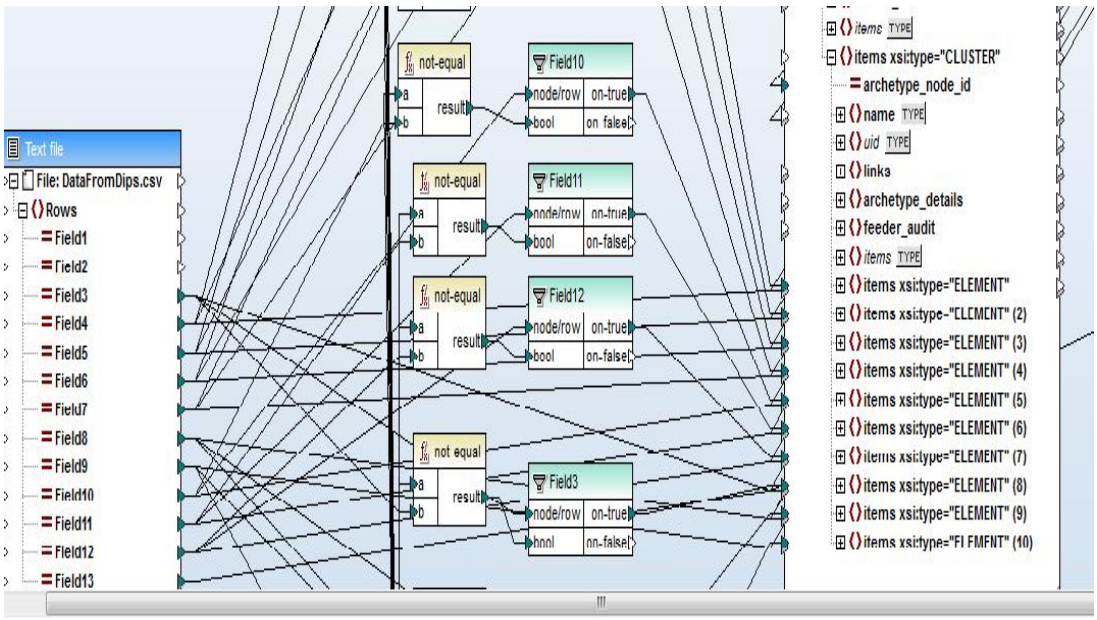


Figure 6.2: CSV to Composition based on flat template mapping.

The second important component of the application was RULE-2 and it was implemented similarly using the mapping tool, but in this case the mapping was done between XML files. This module corresponds to the annotate use case. When a user click on annotate, we queried all the compositions in each EHR committed and the result of the query was a single XML file as shown in the Figure 6.3 containing a list of N dynamic number of compositions. We mapped the XML result to N number of XML composition files. The compositions in the queried XML file were based on the flat template while the target compositions were validated using the template referencing the design archetypes.

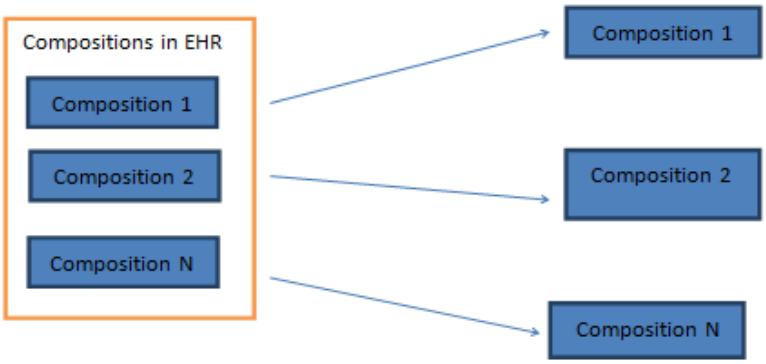


Figure 6.3: mapping the queried XML to N different compositions XML.

Altova© Mapforce supports code generation for mapping XML file into multiple dynamic number of XML files and we used that functionality. Figure 6.4 shows how we mapped the XML schemas to accomplish this task.

The source code was generated in C# and we imported the generated project as an existing project to our project. Using this project we generated XML compositions and each XML composition file was deserialized as RM object and committed to the remote server. We have included a sample XML file which is the output from RULE-2 at appendix B.

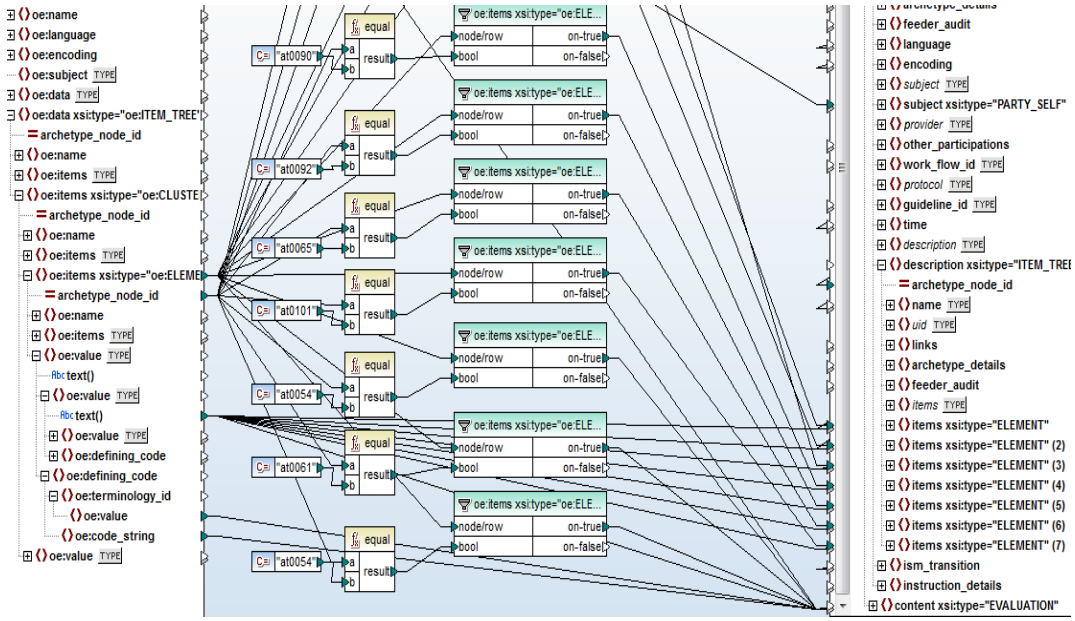


Figure 6.4: mapping between the queried XML to composition based on template using design archetypes.

After importing the project to our application, we used the code shown below to start the mapping process.

```
using Mapping;
using Altova.Types;
MappingMapToComposition MappingMapToCompositionObject = new MappingMapToComposition();
string file1 = @"Queryresult.xml";
string file2 = @"Rule2.csv";
string path1 =Path.GetFullPath(file1);
string path2= Path.GetFullPath(file2);
Altova.IO.Input xmlresult2Source = Altova.IO.StreamInput.createInput(path1);
Altova.IO.Input MapRule2v2Source = Altova.IO.StreamInput.createInput(path2);
MappingMapToCompositionObject.Run( xmlresult2Source, MapRule2v2Source);
```

The user interface and visualization of the data were implemented using a windows form. As the overall aim of this thesis was assessing the feasibility of the platform, we did not focus on visualization; better presentation of data can be done using XSLT that transform XML query result to HTML. We implemented this use case by sending fine grained AQL queries at an element level in parallel to the server and visualizing the result using a windows form. We have attached the snapshot of the application at appendix

C.

6.1 Summary

The chapter starts by detailing the specifics of the programming environment used. Then the chapter details how the data was committed to the oceans openEHR server, and queried back to get the committed data. The chapter also explains what the input and out of the mapper modules were and how they were implemented using the mapping tool. Then we discussed how the source code generated was integrated with our application.

Chapter 7

Result and Discussion

This chapter deals with all the main findings and the possible interpretations of such findings. We will discuss the feasibility of using the solution based on the result obtained from the prototype.

7.1 Result

In this study, we designed a flat archetype and template for committing the extracted data. For this purpose, openEHR provides Entry called GENERIC_ENTRY. We generated XML compositions using GENERIC_ENTRY. We were able to validate the XML successfully using the XML validation tool, but we could not build the RM object using the OpenEHR project; as a result, EVALUATION Entry was used for demonstration. However, future implementation should consider using GENERIC_ENTRY as it is the Entry designed for that purpose. Both Entry has the same mandatory variable called 'data' and we used this variable to represent the structure of the extracted data. The main idea behind using flat archetype is to design archetypes that mimic the structure of the extracted data; designing this archetype does not need medical knowledge. Figure

7.1 shows the generic template designed using the flat archetype made with the EVALUATION Entry. Each field in the CSV file was mapped with ELEMENT in the flat archetype.

While saving the extracted data as a composition instance, it was validated using a template referencing the flat archetype at the server side before committing. At this stage, we had changed the format of the extracted data to openEHR format. We queried the result back and visualize the data saved.

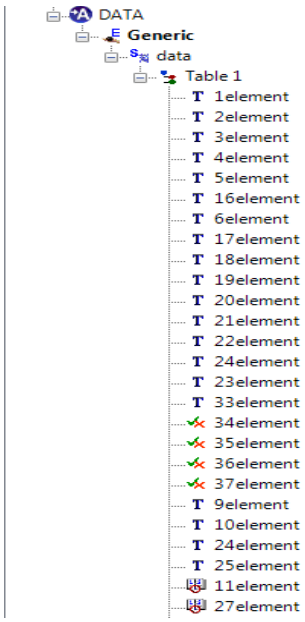


Figure 7.1: flat template.

As a part of this project, we had designed another template, referencing design archetypes. The design archetypes were downloaded from the CKM. These archetypes were made by clinicians and follow ontological design. The template made using the design archetype is shown in Figure 7.3.

We queried all the compositions that conform to the template, referencing the flat archetype for each EHR. Then we transformed them to other compositions that conform to the template, referencing the design archetypes. These compositions can be

easily annotated with semantic information by using the design archetypes as a gateway to external terminology. A study has proved that annotation using design archetypes is easier than traditional annotation and Figure 7.2 shows how archetype can be used to annotate data easily (Bisbal et al. 2010).

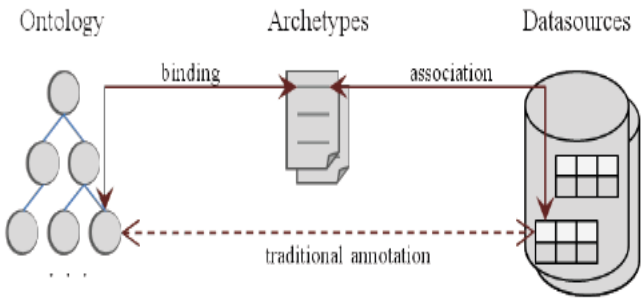


Figure 7.2: annotation with archetype (Bisbal et al. 2010)

However; the efficiency of the developed prototype was affected by connection setup time and net-work delay between the client running EhrGate.WSClient and the server every time we commit and query data. In order to upgrade the performance of the system, we created a thread pool to send each query to the server in parallel. We visualized the result of the committed data using windows form.

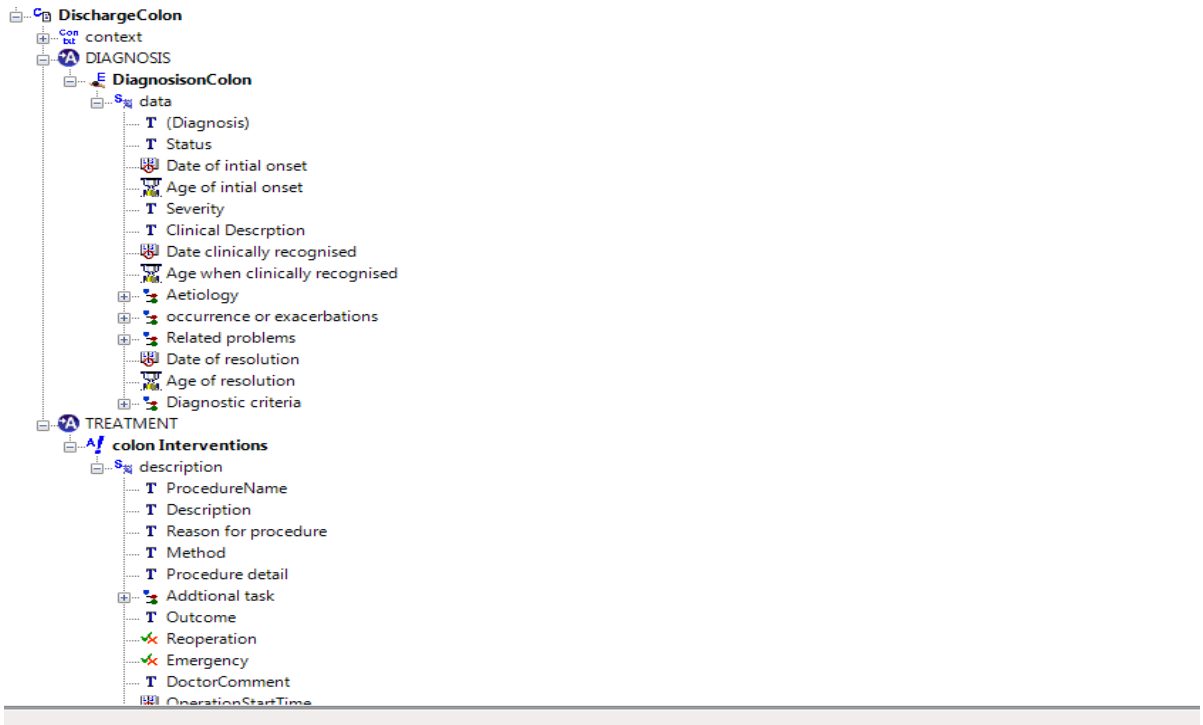


Figure 7.3: template using designed archetype

7.1.1 Feasibility analysis

In the following section, we will discuss the feasibility of the solution based on the criteria stated in the method section.

7.1.1.1 Expected value gain

One of the problems with the current research repositories was that the quality of the data were not up to research standards. In an openEHR based repository, the data were validated using the template before committing, and this can upgrade the quality of data in the repository. The templates were based on archetypes created by healthcare workers. The archetypes' field can be defined by clinicians to the type of data appropriate for that concept; as a result the data committed to the repository were always valid with respect

to the data type defined by clinicians.

When compared with our electronic health record, openEHR provides more data types. There are concepts in openEHR which are not defined in our electronic health record. For example concepts like duration were not defined in our EHR and for the defined concepts like DATE, we had different option to map to including; *date and time*, *date only* or *time only* option in openEHR. Thus we found openEHR to be more expressive. This variety of data types will also help to precisely express the data to be recorded in a specific field. This property can help to upgrade the quality of data in the repository by excluding other similar data types easily. Moreover, it clearly makes the data more structured and computer processable.

In addition to the above advantages archetype enabled research repositories allow fine grained semantic queries and this can help in retrieving more tailored information to the researchers need. Researchers normally want to do statistical or mathematical analysis on the data. For instance, researchers can use semantic queries to retrieve all patients' age where re-operation has been carried out. On the result, they can easily do statistical and mathematical operations like getting the mean, variance, and mode of the retrieved result to further analyse.

Archetype enabled research repositories allow binding archetypes with terminology (Qamar 2008) and this will help to standardize the data. The archetype approach supports achieving highly interoperable system since it is based on a standardised reference model, a standardized service interface model, and enable standardized domain specific concept by sharing archetypes. Archetypes data is more structured and standardized, as a result automatic processing can easily be done on the data using computing systems.

The prototype have met the collected functional and non-functional requirements. As a result, we expect a high value gain from using the approach and this criterion was given a "high" value.

7.1.1.2 Resources

We used clinical modelling tools from ocean informatics and these include; archetype editor, template designer and ADL workbench. They are easy to use because of the drag and drop tools which can be freely downloaded from oceans informatics web-page. These clinical modelling tools helped us a lot while developing the application. However, there was a lack of documentation which can prevent developers from using the tools effectively.

We used Altova© Mapforce visual XML mapping tool to map data and generate transformation rules; it is a commercial product that needs to be purchased. There were also other similar solutions from other vendors. We found the Altova© Mapforce tool and the documentation helpful in the prototype development process.

We used OceanEHR to save the extracted data; it is a secure and high-performance health record computing framework. The project OpenEHR.NET and EhrGate.WSClient were deployed on the client side and helped us to ease the application development task. We used a proprietary C# implemented openEHR platform called OceanEHR, but there is also open source Java implemented openEHR platform which can be accessed freely (Chen & Klein 2007).

We found openEHR central repository of freely available archetypes or the clinical knowledge manger to be a valuable source of archetypes. By re-using existing high quality archetype designs, we were able to significantly speed up the time it took to design and develop an archetype based system. openEHR technical discussion group was also a vital source of information when we got into technical difficulties. In this study, we were able to access all the required resources to build the prototype; as a result, this criterion was given "high" value.

7.1.1.3 Complexity

We observed that the process of writing the translation code for transforming the extracted CSV or the queried XML file to XML file was labour intensive; as a result, a more efficient approach would be to automate the writing of the translation code. We used a tool which generates code based on the visual mappings between the source and target data structures. Though the tool helped us to map the data easily, we found mapping between big schemas to be vulnerable to errors since it is easy to inadvertently delete the connection made between the schema elements. However, effective use the XML mapping could simplify this problem.

The process of annotating the data was done using design archetypes and this involved mapping the queried XML file to another composition XML file based on the design archetypes. Every time the clinicians need to annotate the data mapping rule/ code also has to be modified. These could add complexity to the system. Using our experience as a reference, and the reasons mentioned above, we gave "medium" to the criteria complexity of implementing the system.

7.1.1.4 Scope

This criterion classifies whether the system can be developed by applying only internal resources, external resources of a system implementer, or both. It is known that managing external resource is more difficult than managing internal resource. The value given were

- Internal only (developer time and knowledge, tools and hardware)
- Internal and external
- External only

While building the prototype in additional to internal resources, external resources were applied. The project involves strong collaboration with clinicians. The process of selecting relevant archetypes for the application and designing new archetypes when needed, requires participation of healthcare workers. In addition to that while developing the prototype, we used an external and remote server to save the data. As a result this criterion was given the value of "Internal and external".

7.1.1.5 Development time frame

Based on the prototype development experience factors that will affect the application development time are the processes of analysing the data, selecting and designing the archetypes appropriate for the application. Especially archetypes for research repositories need to be specialized so as to add more relevant information relevant for the researchers. Annotating the data via archetype binding with terminology can also take time, though this approach was found to be simple when compared to the traditional annotation approach (Bisbal et al. 2010). Due to these reasons this criterion was given value of "medium". Figure 7.4 summarizes our discussion on the feasibility of the solution.

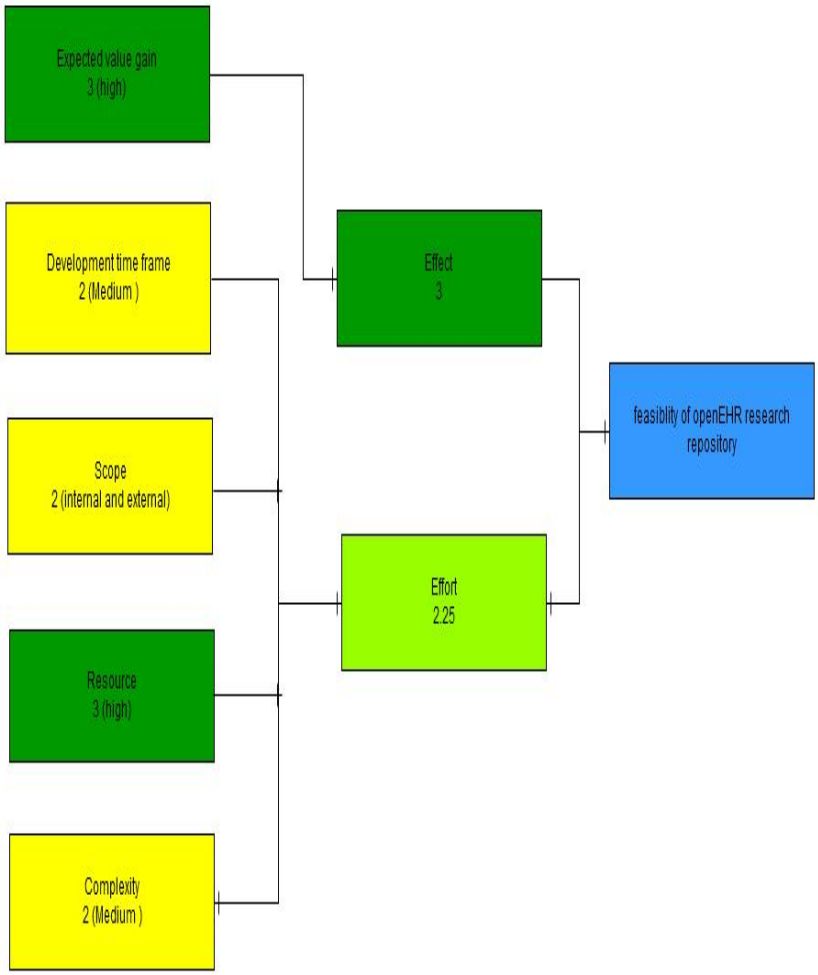


Figure 7.4: feasibility study result

7.2 Discussion

The presented prototype application based on openEHR platform was aimed at building and assessing the feasibility of an approach that could be used to build a research repository.

Before implementing the prototype, we analysed the alternative solutions that could be used for building repository. The other alternative standards to implement the research

repository were HL7 and CEN. Both standards tend to be more focused on communicating the patient's information between applications rather than on building data model that can be used for storing patient data. As a result, we found openEHR to be a suitable candidate for implementing the research repository.

The implemented prototype showed that openEHR has a high potential to build a research repository. Stakeholders involved including; the researchers, clinicians, patients and technical persons could benefit a lot from using the system.

In the archetype based approach the clinicians are not passive, but actively determine the possible breadth, depth, and richness of the research data kept in the repository. They directly affect the quality of the data in the repository through their role in creating the design archetypes. The design archetypes are used as a gateway to annotate the data in the repository. Empowering the clinician is important since they have the knowledge to expanded the extracted data using archetypes so as to enable a high quality research. This major role they play could motivate them to participate actively in the research repository building process.

Researchers can benefit from the system as the archetype methodology enables to have a standardized and structured data. Furthermore; archetypes provide the basis for powerful searching or querying EHR research repositories so as to retrieve more relevant and precise information based on the researchers need. Patients are also the major beneficiary from the system as the outcome research repository enables a continuous assessment of the treatment procedures and guidelines used to treat patients. This will improve the quality of care being given be the healthcare institution.

Technical persons are needed to extract and map the data in electronic health record to the openEHR platform using flat archetypes. Generating the transformation code to the view, based on design archetypes produced by health care workers, is also done by technical persons. In order to accomplish these tasks, we used different openEHR

clinical modelling tools and most of the time these tools reduce openEHR application development task to XML programming. Technical persons can also benefit from this since currently XML is a matured technology and developers have enough knowledge to work with this technology. Moreover, there are different tools that can be used for working with XML files. As a result, technical persons can develop and maintain the application without worrying much about openEHR technical details.

7.3 Summary

In this chapter we have discussed the result and possible interpretation of our result. We have also outlined the overall feasibility of openEHR as a research repository using the method we developed for assessing the feasibility. The value assignment to the feasibility criterion was done based on our experience on developing the solution; as a result, there could be subjectivity. However, the project gave us valuable input about the feasibility of the system.

Chapter 8

Conclusion and future work

8.1 Conclusion

The problem statement of our research was:

"Is it feasible to use openEHR to make the data in the routinely collected DIPS electronic health record available for research and outcome assessment? "

This problem was answered by building a research repository using an openEHR implemented platform and assessing the feasibility of the developed system using a criterion. The criterion assess the feasibility by measuring the effect of the system and effort needed to build the repository. The overall feasibility of developing a research repository using openEHR platform showed high potential effect and medium effort attached to it. The system provides a standardized and structured high quality data to enable outcome research on data collected from EHRs. Developing the system in addition to internal resource needs a strong collaboration with healthcare workers. The process of annotating and expanding the data using archetypes could also need some time, but this approach is easier when compared with traditional methods of data annotating. Since

the result of our study was based on the assessment of our development experience, to strengthen the evidence further research needs to be done by developers with different background.

8.2 Contributions

Solution to demonstrate secondary use of EHR data: Even though it is widely known that secondary use of EHR data can expand knowledge about disease and appropriate treatments, there has been little research to support this claim (Hersh 2007) (Gall et al. 2011). Our study fills some of this gap by providing a solution that allows demonstrating the value of secondary use of EHR data.

Assessing openEHR feasibility: While we were doing literature review to assess the current situation in this area, we found a number of studies claiming that openEHR can be used for building research repository. However, we found few studies that have implemented research repository using this platform and assessed the feasibility. This study contributes to this area by filling some of this gap.

8.3 Future work

Though we have used a limited data set, the prototype gave us useful information about the feasibility of the approach to build research repositories. Future studies can study the applicability of the system with a wide scoped data set.

For visualizing the data in the repository, we had used windows form and it was filled with parallel queries result at element value level, but future implementation should consider using technologies like XForm which can be used for visualizing the XML query result. Querying the data at higher level (archetype level) and visualizing that result

could also improve the efficiency of the system. Moreover, studies can also include focus on extracting the committed data in file formats appropriate for statistical processing tools used for research purpose.

It is known that data for research purpose need to be elaborated with additional information important for researchers and this can be done in openEHR with help of specialized archetypes. In our study, we used archetypes from the clinical knowledge manger, but interesting feedback can be gathered by specializing and bounding these archetypes with external terminology. Future work can also consider using that and studying the operational feasibility of the system.

Appendix A

composition generated with RULE-1

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<composition archetype_node_id="openEHR-EHR-COMPOSITION.tempgeneric.v1" xmlns:xsi="http://www.w3.org/2001/XMLSchem
xsi:schemaLocation="http://schemas.openehr.org/v1 C:/Users/leykun/Documents/project/Composition.xsd" xmlns="http://s
    <name>
        <value>TempGeneric</value>
    </name>
    <archetype_details>
        <archetype_id>
            <value>openEHR-EHR-COMPOSITION.tempgeneric.v1</value>
        </archetype_id>
        <template_id>
            <value>TempGeneric</value>
        </template_id>
        <rm_version>1.0.1</rm_version>
    </archetype_details>
    <language>
        <terminology_id>
            <value>ISO_639-1</value>
        </terminology_id>
        <code_string>en</code_string>
    </language>
    <territory>
        <terminology_id>
            <value>ISO_3166-1</value>
        </terminology_id>
        <code_string>AU</code_string>
    </territory>
```

```
<category>
        <value>event</value>
        <defining\_code>
                <terminology\_id>
                        <value>openehr</value>
                </terminology\_id>
                <code\_string>433</code\_string>
        </defining\_code>
</category>
<composer xmlns:n1="http://schemas.openehr.org/v1" xsi:type="n1:PARTY\_IDENTIFIED">
        <name>EhrGateUnit</name>
</composer>
<content xmlns:n1="http://schemas.openehr.org/v1" archetype\_node\_id="openEHR-EHR-EVALUATION.temgeneric.v1"
        <name>
                <value>Diagnosis</value>
        </name>
        <language>
                <terminology\_id>
                        <value>ISO\_639-1</value>
                </terminology\_id>
                <code\_string>en</code\_string>
        </language>
        <encoding>
                <terminology\_id>
                        <value>IANA\_character-sets</value>
                </terminology\_id>
                <code\_string>UTF-8</code\_string>
        </encoding>
        <subject xsi:type="n1:PARTY\_SELF"/>
        <data archetype\_node\_id="at0001" xsi:type="n1:ITEM\_TREE">
                <name>
                        <value>Tree</value>
                </name>
                <items archetype\_node\_id="at0031" xsi:type="n1:CLUSTER">
                        <name>
                                <value>Table 1</value>
                        </name>
                        <items archetype\_node\_id="at0055" xsi:type="n1:ELEMENT">
                                <name>
                                        <value>3element</value>
                                </name>
                                <value xsi:type="n1:DV\_TEXT">
                                        <value>JFF30</value>
                                </value>
                        </items>
                        <items archetype\_node\_id="at0056" xsi:type="n1:ELEMENT">
                                <name>
                                        <value>4element</value>
                                </name>
                                <value xsi:type="n1:DV\_TEXT">
                                        <value>Annen kolostomi</value>
                                </value>
```

```
                        </items>
                        <items archetype_node_id="at0090" xsi:type="n1:ELEMENT">
                                <name>
                                        <value>34element</value>
                                </name>
                                <value xsi:type="n1:DV_BOOLEAN">
                                        <value>false</value>
                                </value>
                        </items>
                        <items archetype_node_id="at0092" xsi:type="n1:ELEMENT">
                                <name>
                                        <value>35element</value>
                                </name>
                                <value xsi:type="n1:DV_BOOLEAN">
                                        <value>false</value>
                                </value>
                        </items>
                        <items archetype_node_id="at0061" xsi:type="n1:ELEMENT">
                                <name>
                                        <value>9element</value>
                                </name>
                                <value xsi:type="n1:DV_CODED_TEXT">
                                        <value>Ascending colon</value>
                                        <defining_code>
                                                <terminology_id>
                                                        <value>ICD10</value>
                                                </terminology_id>
                                                <code_string>C182</code_string>
                                        </defining_code>
                                </value>
                        </items>
                        <items archetype_node_id="at0065" xsi:type="n1:ELEMENT">
                                <name>
                                        <value>11element</value>
                                </name>
                                <value xsi:type="n1:DV_DATE">
                                        <value>20060417</value>
                                </value>
                        </items>
                </data>
        </content>
</composition>
```

Appendix B

composition generated with RULE-2

```
<?xml version="1.0" encoding="utf-8"?>
<composition xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://schemas.openehr.org/v1
        <name>
                <value>TestOnColon</value>
        </name>
        <archetype_details>
                <archetype_id>
                        <value>openEHR-EHR-COMPOSITION.testdischarge.v1</value>
                </archetype_id>
                <template_id>
                        <value>TestOnColon</value>
                </template_id>
                <rm_version>1.0.1</rm_version>
        </archetype_details>
        <language>
                <terminology_id>
                        <value>ISO_639-1</value>
                </terminology_id>
                <code_string>en</code_string>
        </language>
        <territory>
                <terminology_id>
                        <value>ISO_3166-1</value>
                </terminology_id>
                <code_string>AU</code_string>
```

```
</territory>
<category>
        <value>event</value>
        <defining\_code>
                <terminology\_id>
                        <value>openehr</value>
                </terminology\_id>
                <code\_string>433</code\_string>
        </defining\_code>
</category>
<composer xsi:type="PARTY\_IDENTIFIED">
        <name>EhrGateUnit</name>
</composer>
<context>
        <start\_time>
                <value>20080102T042746,0468Z</value>
        </start\_time>
        <setting>
                <value>primary medical care</value>
                <defining\_code>
                        <terminology\_id>
                                <value>openehr</value>
                        </terminology\_id>
                        <code\_string>228</code\_string>
                </defining\_code>
        </setting>
</context>
<content xsi:type="ACTION" archetype\_node\_id="openEHR-EHR-ACTION.colonoperation.v1">
        <name>
                <value>colon Interventions</value>
        </name>
        <language>
                <terminology\_id>
                        <value>ISO\_639-1</value>
                </terminology\_id>
                <code\_string>en</code\_string>
        </language>
        <encoding>
                <terminology\_id>
                        <value>IANA\_character-sets</value>
                </terminology\_id>
                <code\_string>UTF-8</code\_string>
        </encoding>
        <subject xsi:type="PARTY\_SELF" />
        <time>
                <value>2012-03-08T15:53Z</value>
        </time>
        <description xsi:type="ITEM\_TREE" archetype\_node\_id="at0001">
                <name>
                        <value>Tree</value>
                </name>
                <items xsi:type="ELEMENT" archetype\_node\_id="at0003">
```

```
                <name>
                        <value>ProcedureName</value>
                </name>
                <value xsi:type="DV\_TEXT">
                        <value>13</value>
                </value>
        </items>
        <items xsi:type="ELEMENT" archetype\_node\_id="at0004">
                <name>
                        <value>Description</value>
                </name>
                <value xsi:type="DV\_TEXT">
                        <value>Informasjon/instruksjon til bruker</value>
                </value>
        </items>
        <items xsi:type="ELEMENT" archetype\_node\_id="at0012">
                <name>
                        <value>Reoperation</value>
                </name>
                <value xsi:type="DV\_BOOLEAN">
                        <value>false</value>
                </value>
        </items>
        <items xsi:type="ELEMENT" archetype\_node\_id="at0013">
                <name>
                        <value>Emergency</value>
                </name>
                <value xsi:type="DV\_BOOLEAN">
                        <value>false</value>
                </value>
        </items>
        <items xsi:type="ELEMENT" archetype\_node\_id="at0017">
                <name>
                        <value>StartTime</value>
                </name>
                <value xsi:type="DV\_DATE">
                        <value>20030714</value>
                </value>
        </items>
        <items xsi:type="ELEMENT" archetype\_node\_id="at0018">
                <name>
                        <value>EndTime</value>
                </name>
                <value xsi:type="DV\_DATE">
                        <value>20030714</value>
                </value>
        </items>
</description>
<ism\_transition>
        <current\_state>
                <value>Completed</value>
                <defining\_code>
```

```
                                        <terminology\_id>
                                                <value>openehr</value>
                                        </terminology\_id>
                                        <code\_string>532</code\_string>
                                </defining\_code>
                        </current\_state>
                </ism\_transition>
        </content>
        <content xsi:type="EVALUATION" archetype\_node\_id="openEHR-EHR-EVALUATION.testcolon.v1">
                <name>
                        <value>DiagnosisonColon</value>
                </name>
                <language>
                        <terminology\_id>
                                <value>ISO\_639-1</value>
                        </terminology\_id>
                        <code\_string>en</code\_string>
                </language>
                <encoding>
                        <terminology\_id>
                                <value>IANA\_character-sets</value>
                        </terminology\_id>
                        <code\_string>UTF-8</code\_string>
                </encoding>
                <subject xsi:type="PARTY\_SELF" />
                <data xsi:type="ITEM\_TREE" archetype\_node\_id="at0001">
                        <name>
                                <value>Tree</value>
                        </name>
                        <items xsi:type="ELEMENT" archetype\_node\_id="at0002">
                                <name>
                                        <value>Diagnosis</value>
                                </name>
                                <value xsi:type="DV\_CODED\_TEXT">
                                        <value>Ascending colon</value>
                                        <defining\_code>
                                                <terminology\_id>
                                                        <value>ICD10</value>
                                                </terminology\_id>
                                                <code\_string>C182</code\_string>
                                        </defining\_code>
                                </value>
                        </items>
                </data>
        </content>
</composition>
```

Appendix C

snapshot

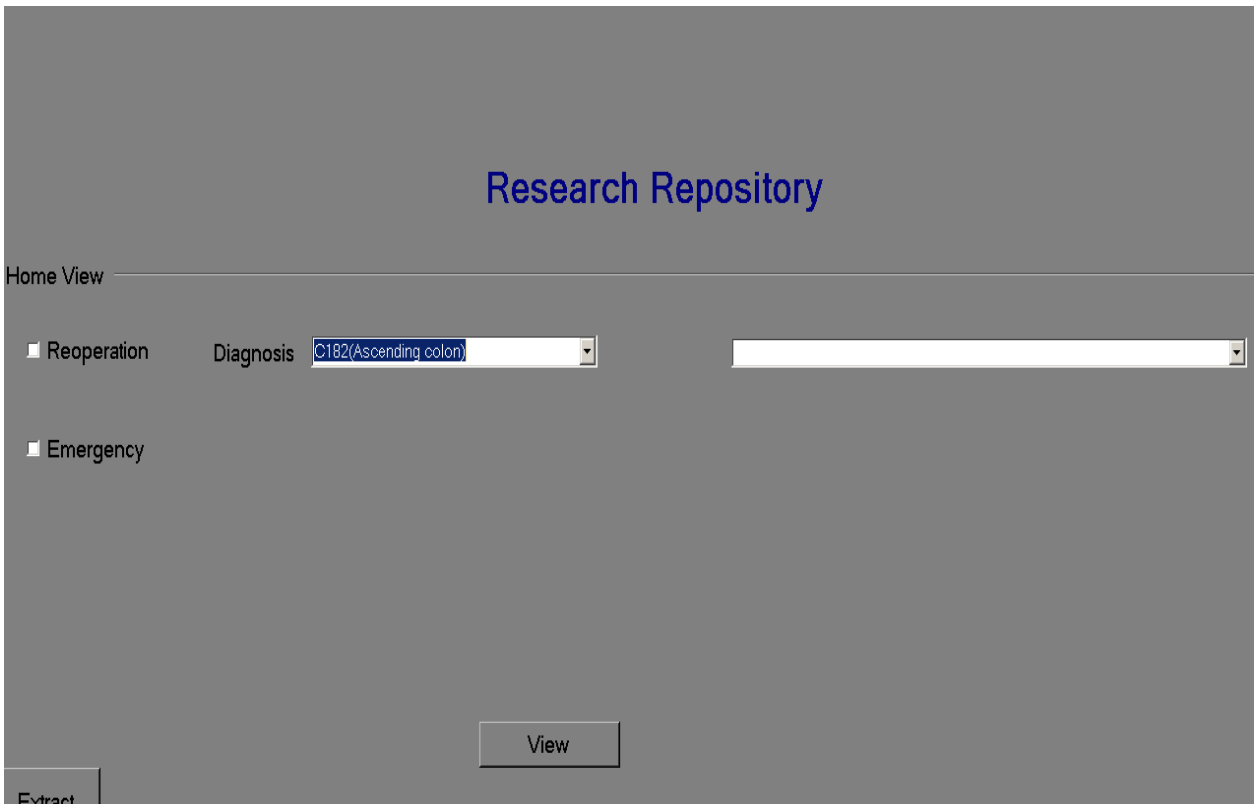


Figure C.1: Home view.

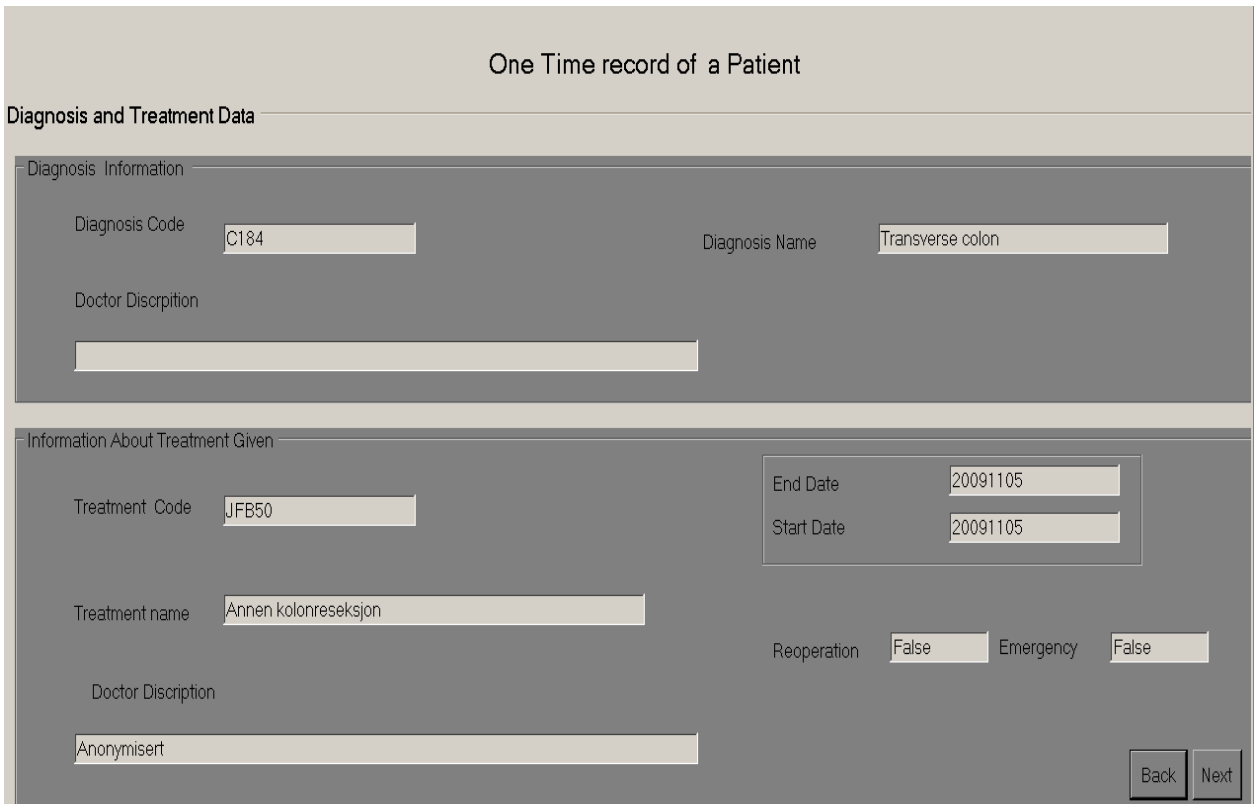


Figure C.2: annotated view.

Bibliography

Anani, N., Chen, R. & Koch, S. (2011), 'Exchanging data via openehr for patient-specific overviews in intensive care: A case study'.

Atalag, K. (2007), Archetype based domain modeling for health information systems, PhD thesis, Middle East Technical University.

Beale, T. (2002), Archetypes: Constraint-based domain models for future-proof information systems, *in* 'OOPSLA 2002 workshop on behavioural semantics', pp. 1–18.

Beale, T. (2003), 'Archetypes and the ehr', *Studies in Health Technology and Informatics* pp. 238–246.

Beale, T. (2005), 'The openehr integration information model, rev. 0.5. openehr reference model', *The openEHR Foundation* .

Beale, T. (2008), 'The openehr archetype model: Archetype object model', *the openEHR release* **1**(2).

Beale, T. & Heard, S. (2003), 'The openehr ehr service model', *Revision 0.2. openEHR Reference Model, the openEHR foundation* .

Beale, T. & Heard, S. (2005), 'Archetype definitions and principles', *Revision 0.6), March* .

Beale, T., Heard, S., Kalra, D. & Lloyd, D. (2006*a*), 'Openehr architecture overview', *The OpenEHR Foundation* .

Beale, T., Heard, S., Kalra, D. & Lloyd, D. (2006*b*), 'Reference model: The openehr ehr information model', *openEHR Specification Release* **1**.

Begoyan, A. (2007), 'An overview of interoperability standards for electronic health records', *USA: society for design and process science* .

Berges, I., Bermudez, J., Goñi, A. & Illarramendi, A. (2010), Semantic interoperability of clinical data, *in* 'Proceedings of the First International Workshop on Model-Driven Interoperability', ACM, pp. 10–14.

Berges, I., Bermudez, J. & Illarramendi, A. (2011), 'Towards semantic interoperability of electronic health records', *Information Technology in Biomedicine, IEEE Transactions on* **9**(99), 10–14.

Bird, L., Goodchild, A. & Tun, Z. (2003), 'Experiences with a two-level modelling approach to electronic health records', *Journal of Research and Practice in Information Technology* **35**(2), 121–138.

Bisbal, J., Engelbrecht, G. & Frangi, A. (2010), Archetype-based semantic mediation: Incremental provisioning of data services, *in* 'Computer-Based Medical Systems (CBMS), 2010 IEEE 23rd International Symposium on', IEEE, pp. 132–137.

Chassin, M. (1996), 'Improving the quality of care', *New England Journal of Medicine* **335**(14), 1060–1063.

Chen, R. (2009), *Towards Interoperable and Knowledge-Based Electronic Health Records Using Archetype Methodology*, Linköping University Electronic Press,.

Chen, R. & Enberg, G. (2007), 'Ge and klein, go (2007) julius a template based supplementary electronic health record system', *BMC Medical Informatics and Decision Making* .

Chen, R., Gunnar, K., Erik, S., Daniel, K. & Hans, Å. (2009), 'Archetype-based conversion of ehr content models: pilot experience with a regional ehr system', *BMC Medical Informatics and Decision Making* **9**.

Chen, R. & Klein, G. (2007), 'The openehr java reference implementation project', *Studies in health technology and informatics* **129**(1), 58.

Coiera, E. (2003), *Guide to health informatics*, Vol. 4, Arnold London, UK.

Comer, D., Gries, D., Mulder, M., Tucker, A., Turner, A., Young, P. & Denning, P. (1989), 'Computing as a discipline', *Communications of the ACM* **32**(1), 9–23.

Corrigan, D. (2010), 'Towards use of openehr archetypes to support views of cystic fibrosis review records'.

Courtney, B. (2010), 'An investigation into the use of hl7 clinical document architecture as a standard for discharge summaries in ireland', *Centre for Health Informatics, Trinity College Dublin* .

Dogac, A., Namli, T., Okcan, A., Laleci, G., Kabak, Y. & Eichelberg, M. (2007), 'Key issues of technical interoperability solutions in ehealth and the ride project', *Software R&D Center, Dept. of Computer Eng., Middle East Technical University, Ankara* **6531**.

Dolin, R., Alschuler, L., Beebe, C., Biron, P., Boyer, S., Essin, D., Kimber, E., Lincoln, T. & Mattison, J. (2001), 'The hl7 clinical document architecture', *Journal of the American Medical Informatics Association* **8**(6), 552–569.

Eggebraaten, T., Tenner, J. & Dubbels, J. (2007), 'A health-care data model based on the hl7 reference information model', *IBM Systems Journal* **46**(1), 5–18.

Eichelberg, M., Aden, T., Riesmeier, J., Dogac, A. & Laleci, G. (2005), 'A survey and analysis of electronic healthcare record standards', *ACM Computing Surveys (CSUR)* **37**(4), 277–315.

Embi, P., Kaufman, S. & Payne, P. (2009), 'Biomedical informatics and outcomes research', *Circulation* **120**(23), 2393–2399.

FDA, U. (1997), 'Design control guidance for medical device manufacturers', *March* **11**, 5.

Gall, M., Grechenig, T. & Bjerre, M. (2011), 'Assessing the feasibility of developing a federated erp system', *Arxiv preprint arXiv:1109.0098* .

Garde, S., Hullin, C., Chen, R., Schuler, T., Granz, J., Knaup, P. & Hovenga, E. (2007), 'Towards sustainability of health information systems: how can we define, measure and achieve it?', *Studies in Health Technology and Informatics* **129**(2), 1179.

Garde, S., Knaup, P., Hovenga, E. & Heard, S. (2007), 'Towards semantic interoperability for electronic health records–domain knowledge governance for open ehr archetypes', *Methods of information in medicine* **46**(3), 332–343.

Garde, S., Knaup, P., Schuler, T. & Hovenga, E. (2005), 'Can openehr archetypes empower multi-centre clinical research?', *Studies in health technology and informatics* **116**, 971–976.

Gok, M. (2008), Introducing an openEHR-based electronic health record system in a hospital, PhD thesis, Master Thesis.

Gruber, T. et al. (1993), 'A translation approach to portable ontology specifications', *Knowledge acquisition* **5**(2), 199–220.

Gunther Schadow, C. (2006), The hl7 reference information model under scrutiny, *in* 'Ubiquity: technologies for better health in aging societies: proceedings of MIE2006', Vol. 124, Ios Pr Inc, p. 151.

Heard, S., Beale, T., Mori, A. & Pishec, O. (2003), 'Templates and archetypes: how do we know what we are talking about', *URL: http://www. openehr. org/publications/archetypes/templates_ and_ archetypes_ heard_ et_ al. pdf* .

Heard, S. & Leslie, H. (2008), 'Building an archetype.'.
URL: *http://www.oceaninformatics.com/Media/docs/2008-06-Building-Archetypes-5c9eaef0-62de-47c6-a155-2ff853e0045f.pdf*

Hersh, W. (2007), 'Adding value to the electronic health record through secondary use of data for quality assurance, research, and surveillance', *Clin Pharmacol Ther* **81**, 126–128.

HL7 (2007), 'About health level seven international'.
URL: *http://www.hl7.org/about/index.cfm*

HL7 (2012), 'Health level seven home page'.
URL: *http://www.hl7.org/*

Iakovidis, I. (1998), 'Towards personal health record: current situation, obstacles and trends in implementation of electronic healthcare record in europe1', *International journal of medical informatics* **52**(1-3), 105–115.

Iavindrasana, J., Iacono, L., Muller, H., Periz, I., Summers, P. & Wright, J. (2008), Access to clinical information systems for research in the life sciences: security and privacy considerations, *in* 'Proceedings of HealthGrid'.

IEEE (1998), 'Ieee guide for developing system requirements specifications (ieee std 1233, 1998 edition)'.

informatics, O. (2009), 'Product description'.

URL: *http://www.oceaninformatics.com/media/docs/ProductDescription-a4e6f624-5014-46e2-85f3-e47b7c6a8fb5.pdf*

ISO (2012), 'About iso'.

URL: *http://www.iso.org/iso/about.htm*

Kalra, D. (2006), 'Electronic health record standards', *IMIA yearbook of medical informatics* **2006**, 136–144.

Kalra, D., Beale, T. & Heard, S. (2005), 'The openehr foundation', *Studies in Health Technology and Informatics* **115**, 153–173.

Leslie, H. (2008), 'International developments in openehr archetypes and templates', *HIM J* **37**(1), 38–9.

Leslie, H. & Heard, S. (2006), Archetypes 101, *in* 'HIC'.

Ma, C., Frankel, H., Beale, T. & Heard, S. (2007), 'EHR query language (EQL)-a query language for archetype-based health records', *Studies in health technology and informatics* **129**(1), 397.

Maldonado, J., Moner, D., Boscá, D., Fernández-Breis, J., Angulo, C. & Robles, M. (2009), 'Linkehr-ed: A multi-reference model archetype editor based on formal semantics', *International Journal of Medical Informatics* **78**(8), 559–570.

Martinez-Costa, C., Menárguez-Tortosa, M. & Fernández-Breis, J. (2010), 'An approach for the semantic interoperability of iso en 13606 and openehr archetypes', *Journal of Biomedical Informatics* **43**(5), 736–746.

openEHR (2007), 'openehr home page'.

URL: *http://www.openehr.org/home.html*

Parviainen, P., Tihinen, M., Lormans, M., van Solingen, R., Mate, J. & Silva, A. (2004), 'Requirements engineering: Dealing with the complexity of sociotechnical systems development', *Requirements Engineering for Sociotechnical Systems* **1**.

Qamar, R. (2008), Semantic mapping of clinical model data to biomedical terminologies to facilitate interoperability, PhD thesis, the University of Manchester.

Qamar, R., Kola, J. & Rector, A. (2007), Unambiguous data modeling to ensure higher accuracy term binding to clinical terminologies, *in* 'AMIA Annual Symposium Proceedings', Vol. 2007, American Medical Informatics Association, p. 608.

Qamar, R. & Rector, A. (2007), 'Semantic issues in integrating data from different models to achieve data interoperability', *Studies in Health Technology and Informatics* **129**(1), 674.

Radatz, J., Geraci, A. & Katki, F. (1990), 'Ieee standard glossary of software engineering terminology', *IEEE Standards Board, New York, Standard IEEE std* pp. 610–12.

Razavi, A. (2007), 'Applications of knowledge discovery in quality registries-predicting recurrence of breast cancer and analyzing non-compliance with a clinical guideline', *Linkoeping University Medical Dissertations* **1018**.

Rector, A. et al. (1999), 'Clinical terminology: why is it so hard?', *Methods of information in medicine* **38**(4/5), 239–252.

Rinner, C., Wrba, T. & Duftschmid, G. (2007), 'Publishing relational medical data as cen 13606 archetype compliant ehr extracts using xml technologies'.

Robertson, G., Czerwinski, M. & Churchill, J. (2005), Visualization of mappings between schemas, *in* 'Proceedings of the SIGCHI conference on Human factors in computing systems', ACM, pp. 431–439.

Safran, C., Bloomrosen, M., Hammond, W., Labkoff, S., Markel-Fox, S., Tang, P. & Detmer, D. (2007), 'Toward a national framework for the secondary use of health data: an american medical informatics association white paper', *Journal of the American Medical Informatics Association* **14**(1), 1–9.

Sanroma, M., Mateu, A. & Oliveras, K. (2004), 'Survey of electronic health record standards', *Research Group on Artificial Intelligence (BANZAI), Document from the Research Project K4CARE (IST-2004-026968 K4CARE)* .

Schloeffel, P. (2004), 'Current ehr developments: An australian and international perspective. health informatics new zealand conference (hinz2004'.

Schloeffel, P., Beale, T., Hayworth, G., Heard, S. & Leslie, H. (2006), 'The relationship between cen 13606, hl7, and openehr', *HIC 2006 and HINZ 2006: Proceedings* **24**.

Scott, M. (2004), 'Requirements: An introduction'.
URL: *http://www.ibm.com/developerworks/rational/library/4166.html*

Scott, P. (2009), 'Meeting the challenges of healthcare interoperability', *Healthcare IT Management* **4**(3), 24–25.

serefarikan (2012), 'openehr.net'.
URL: *http://openehr.codeplex.com/documentation*

Shafarman, M. (2012), 'Hl7 version 3 the use case for the reference information model-reference'.
URL: *http://www.hl7.org/Library/General/hl7v3rimusecase.pdf*

Stan, O. (2010), Techniques for ensuring interoperability in an electronic health record. available online.
URL: *http://utcluj.ro/download/doctorat/Rezumat_Ovidiu_Petru_Stan.pdf*

Stroetman, V., Kalra, D., Lewalle, P., Rector, A., Rodrigues, J., Stroetman, K., Surjan, G., Ustun, B., Virtanen, M. & Zanstra, P. (2009), 'Semantic interoperability for better health and safer healthcare [34 pages]'.

Stroetmann, V., Kalra, D., Lewalle, P., Rector, A., Rodrigues, J., Stroetmann, K., Surjan, G., Ustun, B., Virtanen, M. & Zanstra, P. (2009), 'Semantic interoperability for better health and safer healthcare: Deployment and research roadmap for europe'.

TC, I. (2008), 'Health informatics categorial structure for classifications and coding systems of surgical procedures'.

Tolk, A. (2006), What comes after the semantic web-pads implications for the dynamic web, *in* 'Proceedings of the 20th Workshop on Principles of Advanced and Distributed Simulation', IEEE Computer Society, p. 55.

Touitou, Y., Portaluppi, F., Smolensky, M. & Rensing#, L. (2004), 'Ethical principles and standards for the conduct of human and animal biological rhythm research', *Chronobiology international* **21**(1), 161–170.

Van der Lei, J. (2002), 'Closing the loop between clinical practice, research, and education: the potential of electronic patient records', *Methods of information in medicine* **41**(1), 51–54.

Weiner, M. & Embi, P. (2009), 'Toward reuse of clinical data for research and quality improvement: The end of the beginning?', *Annals of internal medicine* **151**(5), 359–360.

Wollersheim, D., Sari, A. & Rahayu, W. (2009), 'Archetype-based electronic health records: a literature review and evaluation of their applicability to health data interoperability and access', *HIM J* **38**(7), 17.

Wrba, T., Dorda, W., Grossmann, W. & Duftschmid, G. (2008), Analysis of ehrs for research, quality management and health politics, *in* 'eHealth beyond the horizon: get IT there: proceedings of MIE2008, the XXIst International Congress of the European Federation for Medical Informatics', Vol. 136, Ios Pr Inc, p. 425.