Georg-August-Universität
Göttingen
Zentrum für Informatik

ISSN 1612-6793
Nummer ZFI-BM-2008-10

Masterarbeit

im Studiengang Angewandte Informatik

Introducing an openEHR-Based Electronic Health Record System in a Hospital

Case Study, Emergency Department, Austin Health, Melbourne

Murat Gök

Abteilung für
Medizinische Informatik
Universitätsmedizin Göttingen

Bachelor- und Masterarbeiten
des Zentrums für Informatik
an der Georg-August-Universität Göttingen

13.05.2008

Georg-August-Universität Göttingen
Zentrum für Informatik

Lotzestraße 16-18
37083 Göttingen
Deutschland

Tel.	+49 (5 51) 39-1 44 14
Fax	+49 (5 51) 39-1 44 15
Email	office@informatik.uni-goettingen.de
WWW	www.informatik.uni-goettingen.de

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Göttingen, den 13.05.2008

Master Thesis

Introducing an openEHR-Based Electronic Health Record System in a Hospital

Case Study, Emergency Department, Austin Health, Melbourne

Murat Gök

2008-05-13

This thesis arose from the collaboration with Austin Health, the Austin Centre for Applied Clinical Informatics (ACACI), the Nursing Informatics Group, the Biomedical Engineering Department, the Emergency Department, the Central Queensland University Health Informatics Research Group, Ocean Informatics, and the Department of Medical Informatics (University of Goettingen).

UNIVERSITÄTSMEDIZIN GÖTTINGEN UMG

Supervised by:
Prof. Dr. med. Otto Rienhoff
Department of Medical Informatics
University of Goettingen (Germany)
and
Dr. sc. hum. Sebastian Garde
Health Informatics Research Group
Central Queensland University / Ocean Informatics
(Australia)

Acknowledgment

I would like to express my gratitude to all those who supported my work. This thesis could not have been written without Prof Dr Otto Rienhoff, who enabled my stay in Australia and supervised my thesis; Dr Sebastian Garde, who also supervised and supported me thoroughly through this project; Janette Gogler and Dr Carola Hullin who answered every question and questioned every answer.
I am grateful for the editing of this thesis by Genevieve Gogler, Lena Libuda, Ilona Seifart, Helene Schäfer, and Julika Wendland.
I would also like to thank Austin Health with the Nursing Informatics Group, the Austin Centre for Applied Clinical Informatics, the IT Department, the Biomedical Engineering Department, the Emergency Department, and Ocean Informatics.

Thank you so much.

Summary

The aim of this thesis is to provide a roadmap for the introduction of an electronic health record system based on the openEHR (http://www.openehr.org) approach for a health service within a public hospital in Australia.
The idea of electronic health records (EHRs) was born approximately 40 years ago [GL96] and consequently several concepts were developed. One of these approaches is the "Good European Health Record" (GEHR) project on which the openEHR Foundation builds.
Over time the openEHR approach has matured, however, there is still a lack of knowledge on how to introduce an openEHR-based system (implementation and migration strategies). To tackle this problem, the thesis gives an overview of the openEHR approach by presenting the history, architecture, and relations to other standards in electronic health care. The patient flow in an emergency department (ED) of a public hospital (Austin Health) is then analysed in regards to the information produced and documented. This thesis investigates how the data items in the ED can be gathered and mapped to openEHR archetypes, thus formally representing the clinical knowledge. The reusable archetypes cover more than 70% of all archetypes needed in the ED. This figure may vary for other departments. It also points at the development of openEHR templates (a combination of archetypes) through utilising mind maps. Using an example of a ventilation system, data can be migrated from proprietary systems and transferred to an openEHR-based data storage. An explanation is given for an openEHR architecture based EHR system, providing the foundation for the implementation of an openEHR-based prototype.
The thesis shows how an openEHR architecture based EHR system can be introduced in practical terms and how this could lead to interoperability within a department.

Contents

Chapter 1

Introduction

1.1 The Electronic Health Record (EHR)

Documentation of patient's diseases and treatments have existed since people with medical skills were able to report them. This concept of records was time oriented and stayed relevant for centuries. In the late 19th century, as expansion of medical knowledge increased, some hospitals introduced patient medical records. These were mostly leather bound ledgers for each physician, laboratory worker, and surgeon. In the beginning of the 20th century, records were introduced for each patient. This step entailed some advantages. Notes were no longer separated in different ledgers in different departments. Therefore, they were easier to handle and access. In the 1920s, the absence of standards for the documentation of data was realized and discussed. The massive expansion of medical knowledge in the 1950s, and no agreement was reached between physicians on how these data should be ordered, led to problems. The records were unordered and complex. Therefore, Weed published his proposals of the problem oriented medical record (POMR) in 1968. In these POMRs, all notes were recorded in the context of a specific problem [vG95].
At the same time, computers proceeded an enormous development. One of the first manifestations of the electronic medical records was the conference for Information Processing of Medical Records held in 1971 (Lyon) [GL96]. The information processing of medical records led to new challenges and great expectations. New approaches arose for the new topic. The first developments were the computer based versions of the TOR, the POMR, and the Regenstrief approach. These early versions were focused on the structure of data [GL96]. In the early 1980s, information systems were introduced in hospitals in some European countries. Structured data was not sufficient for these approaches. They needed functional interoperability for the communication between systems. In 1987, the first Health Level 7 (HL7) draft stan-

dard was prepared[1] and improved the communication. In the early 1990s, the European Union founded the Good European Health Record (GEHR) project. The main results were the distinction of knowledge and information as well as functional and semantic interoperability. They build the basis of the openEHR approach for an electronic health record architecture.

1.2 The Impacts and Problems of EHRs

EHRs play an increasingly important role as a potential contribution of e-health to the delivery of safer, more efficient, better quality health care [neh06, PPTK05]. There are a few major national initiatives involving significant investments in the UK, US, Canada, and some EU countries. The aims of these projects are to increase the patient's safety and reduce the high cost of inappropriate or inadequate health care. The use of more advanced and better integrated clinical systems for clinical decision support, evidence based medicine, and better care coordination in ambulatory, hospital, and long-term care also promises improvements [neh06].
The Center for Information Technology Leadership, based in the US, assessed the value of electronic health care information exchange and interoperability between providers (hospitals and medical group practices) and independent laboratories, radiology centres, and other providers. Based on this, they produced a four level model of interoperability where "Shared EHR" information is seamlessly shared and used by different clinical Software applications throughout the care chain [WPJ$^+$05].

1. **Nonelectronic data** - no use of information technology (IT) to share information (examples: mail, telephone).

2. **Machine-transportable data** - transmission of nonstandardized information via basic IT; manipulation of information within the document is not possible (examples: scanned documents, pictures, portable document format (pdf) files, ...).

3. **Machine-organizable data** - transmission of **structured messages** containing nonstandardized data; for this level the incoming data needs to be translated by interfaces (examples: e-mail of free text, HL-7 messages, ...).

4. **Machine-interpretable data** - transmission of **structured messages** containing **standardized and coded data**; in this level all systems are able to exchange information by using the same formats and vocabularies (examples: automated exchange of coded results from

[1] www.hl7.org.za/patient/httoc.htm

an external lab into a provider's EMR, automated exchange of a patient's "problem list").

According to the study by Walker [WPJ+05], there is a big potential to save money. In the concrete case (the study by Walker) the arrangements of fully standardized health care information exchange and interoperability could yield a net value of $77.8 billion per year once fully implemented.
In the current situation, the health information is largely still caught up in silos and mostly not shareable by clinicians. EHRs are increasingly needed to provide timely, comprehensive, and coordinated healthcare. However breaking up the health information silos to support standardised communication of health information between systems is a difficult process, due to the complexity of the health domain data compared to businesses such as banking. Furthermore, the solutions from other areas do not always adequately fulfil the requirements in the health area [Les]. Information systems in the business domain such as banking have "customers" but they are simplified and abstract versions of a person. "Patients" in clinical systems are anything but this. Their biological and social complexity is manifested directly in clinical information, leading to a far greater challenge than in other domains [Bea05]. The fourth level shows that clinical content needs to be standardised, coded, structured, and machine interpretable. From many research projects, among others the GEHR project, the openEHR Foundation arose, which provides an open electronic health record architecture specification. It is designed to work in partnership with all vendor systems, organisations, and providers to facilitate semantic [Les] and functional interoperability [Sch04] of health information providing a comprehensive and transformational solution. It applies to the capturing and sharing of information from the health domain which holds very complex and dynamic knowledge.

1.3 Motivation

A lot of research has been done and the openEHR Foundation offers open specifications for an EHR architecture providing the basis for interoperability. There are some companies like Ocean Informatics which offer implementations of these concepts and modelling tools. While the benefits of such systems are accepted relatively widely, there is still a lack of knowledge on how to proceed with the introduction of such systems while keeping the department operational and not missing out the development of electronic management of health information (e-health). This includes implementation and migration strategies as well as the overall approach.

1.4 Aims

The overall aim of my thesis is to examine how to introduce an openEHR architecture based EHR system to an emergency department in an Australian metropolitan hospital. Therefore, it provides an exemplary roadmap for the appropriate introduction. In more detail, it is the aim of this thesis to:

1. Provide an analysis of the patient-flow in the ED, especially the information which is documented on paper and electronic based records and systems.

2. Provide a demonstrative roadmap for the introduction of an EHR system based on the openEHR architecture in an emergency department in Australia.

    - Analyse the link between the data items in the ED and the corresponding openEHR archetypes and templates.
    - Provide a concept for the transfer of data (via proprietary interfaces) to an openEHR architecture based data storage.
    - In consequence, providing a concept for introducing an openEHR architecture based EHR system.

1.5 Structure of the Thesis

The thesis is structured in six chapters. Following this introduction, chapter 2 shows the methods used in the study and provides the foundations which reoccur throughout the thesis. It also gives an overview of the openEHR history, architecture, and interoperability aspects in general and to other health technology standards. Afterwards, in chapter 3, the study analyses the Austin Health emergency department with its systems and paper based documentation in regards to the patient flow. Based on the results of the analyses, chapter 4 shows how the clinical processes are mapped by openEHR templates and also provides a closer look to the components of the EHR system. These components are different systems used in the ED with its proprietary interfaces. The thesis will also demonstrate how to migrate the data of the systems via a converter. At the end, chapter 5 shows a foundation of parts of the implementation of the EHR system followed by the discussion of the results in chapter 6.

Chapter 2

Methods and Material

The thesis focuses on a case study of the introduction of an openEHR-based EHR system to an emergency department in a large metropolitan hospital, Austin Health. Austin Health is a major provider of tertiary health services, health professional education, and research in the northeast of Melbourne, Australia. It has 52 services where an EHR system could be introduced.
To present a profile of the introduction, a four step method (figure 2.1) was chosen according to the Waterfall Model[1].

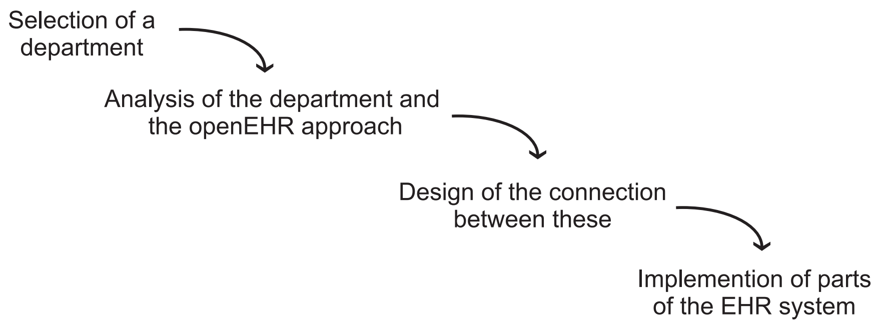

Figure 2.1: Method of study.

In step one, a department was selected which has versatile systems, processes, and gathered data. Therefore, the emergency department (ED) is a very interesting candidate for the introduction of an EHR system. It also covers many aspects which are useful for other departments.
In step two, a literature review was conduced of the openEHR approach including the historical development, the architecture, and the relation to other standards in e-health such as HL7 (CDA), CEN EN 13606, ISO/TR 20514.
For the analysis of the ED, several meetings were performed with an expert. The ED was visited several times and introductions were given of the used

[1]The Waterfall Model is a software development model which is linear and sequential.

systems, processes, and paper based forms. The analysis and tests of the systems were conducted in the biomedical engineering department.
In step three, the connection was designed between the ED, with its different types of information gathering (systems and paper forms), and the openEHR approach. This is feasible through the results of the analysis which shows how the openEHR archetypes match with the documented data items within a clinical service.
Step four was the implementation of significant parts of the developed concepts. These were done by using the programming languages VB.NET and C♯.

In the following a definition of the EHR is given. Afterwards, this chapter introduces foundations which occur throughout the thesis. Section 2.2 introduces the openEHR approach by giving an overview of the history. It outlines the architecture of openEHR, as well as links to other health technology standards. Section 2.3 gives a summary of the Software which is used in this study and provided by Ocean Informatics. Finally, this chapter introduces the Extensible Mark-up Language (XML), the XML Schema, and the XML Path Language (XPath).

2.1 Definition of Electronic Health Records (EHRs)

There are many terms related to the concept of the electronic health records EHR including Computerized Medical Record (CMR), Computerized Patient Record (CPR), Electronic Patient Record (EPR), Electronic Medical Record (EMR), Electronic Health Care Record (EHCR). These terms define, depending on the point of view, many aspects of an EHR. There are a few approaches to give a formal definition, but they "have been unsuccessful due to the difficulty of encapsulating the many and varied facets of the EHR into a single comprehensive definition" [Sch04].
In 2002, the International Standards Organisation (ISO) began to develop an internationally agreed definition of the EHR. One result of this set of standards was the need of distinction between the content and the structure of an EHR. It was also one main result of the earlier conducted Good European Health Record (GEHR) project.
The "basic-generic EHR" defines the simplest ISO definition of an EHR. This definition spans most of the EHRs which are currently in use. Therefore many EHRs do not care about semantic interoperability which is the ability to share and exchange computer processable information within and between health care organisations. This is of fundamental importance to deliver safe and high quality health care in modern, complex, and multidisciplinary health systems [Sch04]. Therefore there is the second definition of the EHR the Integrated Care EHR (ICEHR):

”*The Integrated Care EHR is defined as a repository of information regarding the health of a subject of care in computer processable form, stored and transmitted securely, and accessible by multiple authorised users. It has a commonly agreed logical information model which is independent of EHR systems. Its primary purpose is the support of continuing, efficient and quality integrated health care and it contains information which is retrospective, concurrent and prospective”* [ISO05].

In other words, it is a specialisation of the basic-generic EHR and focuses on the content, privacy, security, and shareability of the EHR. It corresponds to the openEHR definition of an EHR. According to the openEHR website[2] and [GKHH07] an EHR has the following characteristics:

- Patient-centred: one EHR relates to one subject of care, not to an episode of care at an institution.
- Longitudinal: it is a long-term record of care, possibly birth to death.
- Comprehensive: it includes a record of care events from all types of carers and provider institutions tending to a patient, not just one speciality; in other words, there are no important care events of any kind not in the EHR.
- Prospective: not only previous events are recorded, so it is decisional and prospective information such as plans, goals, orders, and evaluations.

It is also important to make a clear distinction between EHR systems and EHRs. An EHR system is defined as a system for recording, retrieving, and manipulating information in electronic health records, or more broadly as the set of components that form the mechanism by which electronic health records are created, used, stored, and retrieved. It includes people, data, rules and procedures, processing and storage devices, and communication and support facilities. In comparison, the EHR itself must be completely independent of the system [Sch04].

2.2 openEHR

The openEHR Foundation - established by the University College London and Ocean Informatics - has developed an electronic health record architecture based on the GEHR - and several other projects of national and international organisations (figure 2.2). It is designed to work in partnership with vendor systems, organisations, and providers to facilitate semantic [Les]

[2] `http://www.openehr.org/shared-resources/openehr_primer.html`

and functional [Sch04, oF07] interoperability of health information.
Therefore, the openEHR architecture is a two level modelling approach for EHRs. The first level of this approach is the reference model; this is a relatively small set of classes used to support the medico-legal requirements and record management functions [GHBK07]. The fist level stands for functional interoperability. It provides the communication between computers like HL7 v2.x [Sch04].
The second level represents the openEHR archetype methodology. Archetypes map clinical knowledge, therefore each archetype represents one clinical concept by constraining instances of the openEHR reference model. This fosters semantic interoperability which is the ability of computers and software to mutually understand the meaning of information. It requires information to be computable, not just shareable [oF07]. The advantage of semantic interoperability is that the shared information is computer processable.
By using two levels - a reference model and archetypes - openEHR separates technical concerns from clinical data collection [GHBK07, oF07, Sch04].

In comparison to ”traditional” single layer systems, where knowledge and information are directly connected, changes are, in a very dynamic domain like the clinical knowledge, very expensive and not easy to arrange [Bea02].

2.2.1 History

The European Union founded a project under the umbrella of the European Health Telematics research programme (Advanced Informatics in Medicine), the GEHR project. It was a three year project, beginning in 1992 and producing public domain documentation of the following: an architecture object model, exchange format sets, and the specifications of access and integration tools for EHR systems. The requirements were the major input into the first ISO EHR standard on requirements for an EHR architecture [CHI05]. After the GEHR project, other projects were conducted in relation to it for further development of the EHR architecture. The most significant ones are listed below in figure 2.2 [SHBR99, Bea07b].

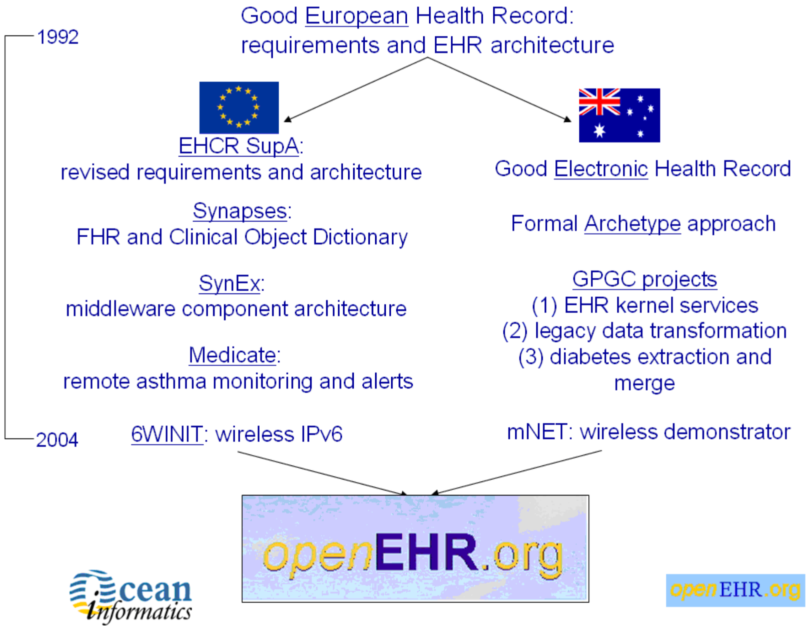

Figure 2.2: History of the openEHR Foundation. Source: [Bea07b].

- EHCR Support Action (EHCR SupA) is a project for re-examining the results of the GEHR Version 1.0 model and other work on EHR systems from around the world. This was carried out with the aim of developing an extended EHR architecture which forms the basis for a new CEN standard.
- Synapses has attempted to develop a 'federated EHR architecture'.
- SynEx had the aims to integrate the work of several European health informatics projects to facilitate the sharing of EHRs between open distributed computing environments through interoperable middle-ware components.
- Medicate was for remote asthma monitoring and alerts.
- 6WINIT validated the introduction of the new Mobile Wireless Internet in Europe[3].

Australia had a review role in the EHCR SupA project and therefore the GEHR is also known as Good Electronic Health Record. In 1997, a small Australian group led by two of the members of the GEHR project, Dr. Sam Heard (from Ocean Informatics) and Tomas Beale (from the University College London [UCL]) extended the GEHR architecture and implemented this

[3]http://www.6winit.org

architecture [SHBR99, Sch04]. They also developed the formal modelling approach, including the archetype methodology. In 2000, they merged the work of UCL and Ocean Informatics to form a new model called openEHR [Sch04]. The openEHR Foundation was created to enable the development of open specifications, software and knowledge resources for health information systems - in particular EHR systems. It publishes all its specifications, and builds reference implementations of them, as open source software. It also develops 'archetypes' and a terminology for uses with EHR's [oF07].

The architecture of the openEHR approach will be introduced next, followed by the relation between different standards in the health sector and openEHR.

2.2.2 openEHR Architecture

This subsection introduces the architecture of the openEHR approach. It describes the two level modelling concepts by examining the reference model, the first level, and the archetype model, the second level. Based on these, it discusses the openEHR templates and the EHR Query Language (EQL).
Figure 2.3 shows the now introduced four levels of information organisation:

1. The reference model is a standardised data representation. It enables interoperability and standardised interfaces.

2. Archetypes represent the clinical knowledge by constraining the reference model.

3. A template is a locally produced constraint specification which articulates which archetypes go together in a screen form or message specification. The EQL stands for standardised querying.

4. The cognitive user interface is a flexible approach to data capture and viewing capability.

The four levels of information structure can be visualised as below.

1. **The Reference Model (RM)** describes the health record itself - not the clinical data that is contained within it [oF07]. According to [BH07a], it contains a few different models like the Support Information Model. It describes the most basic concepts required by all other packages and is comprised of the Definitions, Identification, Terminology, and Measurement packages. These packages are briefly introduced here:

    - **The Data Types Information Model** defines a set of data types and underlies all other models, and provides general as well as clinical types of health information.

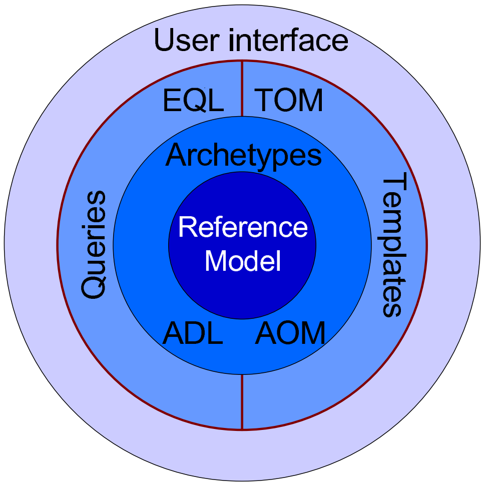

Figure 2.3: Architecture of the openEHR approach. Based on: [Bea07b].

- **The Data Structures Information Model** provides generic data structure such as single items, lists, tables, and trees.
- **The Common Information Model** contains packages for the linkage between archetypes and information.
- **The Security Information Model** defines the semantics of access control and privacy setting for information in the EHR.
- **The EHR Information Model** defines the containment and context semantics of the concepts EHR, COMPOSITION, SECTION, and ENTRY. These classes are the major components of the EHR, and correspond directly with the classes of the same names in CEN EN13606. More information about the interoperability with other standards is given in section 2.2.3.
- **The EHR Extract Information Model** defines how an EHR extract is built from COMPOSITIONs, demographic, and access control information from the EHR. A number of Extract variations are supported, including "full openEHR", and an openEHR/ openEHR synchronisation Extract.
- **The Integration Information Model** defines the class GENERIC_ENTRY which is a subtype of ENTRY. This Entry type has its own archetypes, known as the "integration archetypes", which can be used with clinical archetypes as the basis for a tool-based data integration system.
- **The Demographics Information Model** provides definitions of generic concepts of PARTY, ROLE and related details such as contact addresses.

2. **The Archetype Model (AM)** contains the models which are necessary to describe the semantics of archetypes and templates. These include the Archetype Definition Language (ADL), the archetype object model (AOM), the openEHR archetype profile (OAP), and the template object model (TOM).

   - **The ADL** is a formal language for expressing archetypes, and can be categorised as a knowledge description language [BH07a]. It provides a formal, abstract syntax for describing constraints on any domain entity whose data is described by an information model. It is useful when generic information models are used for representing all data in a system. Archetypes are used to constrain the valid structures of instances of these generic classes to represent the domain concepts [oF07].
   - **The AOM** formalises archetypes. This is an object model of the semantics of archetypes. When an archetype is represented in memory (for example in an archetype-enabled EHR "kernel"), the archetype will exist as an instance of the classes of this model. The AOM is thus the definitive statement of the semantics of archetypes [BH07a]. Archetypes themselves are terminology-neutral, but can link to external terminologies like SNOMED CT [GHBK07]. With regards to their content, archetypes can be classified in three levels [BH07a, Gar]:
     - Descriptive archetypes (entry types)
       * Observation (e.g. blood pressure measurement)
       * Evaluation (e.g. adverse event)
       * Instruction (e.g. medication order)
     - Organisational archetypes
       * Section (document headings)
     - Thematic archetypes
       * Composition (documents)

   Archetypes are a description of valid Entries, Sections, and Compositions. These are expressed in a formal manner which enables them to be shared between systems.
   The oximetry archetype (figure 2.4) for example represents a description of all the information a clinician may wish to report about a measurement, the maximum data set. This archetype can be used in the ED for monitoring purposes.

   Figure 2.5 represents a view of the two layer methodology. It separates the "traditional" approach into "information" and "knowledge". On the information side there is what the name already implies: the information, which structure and semantic is defined by a reference model.

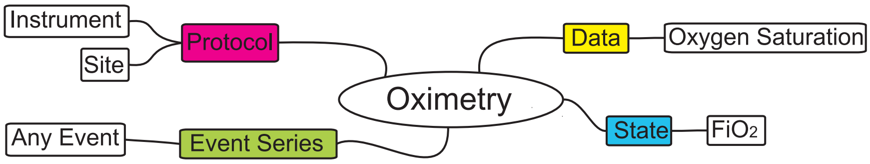

Figure 2.4: Oximetry archetype as mind map.

On the other side, there is the knowledge which is expressed through archetypes. Archetypes are, as introduced earlier, for constraining the structure and semantics of the information model at runtime, in a way that these map clinical concepts. These archetypes are written in the archetype definition language (ADL).
Another very common way to describe this relation is the analogy with LEGO® bricks. One can see that the semantics of the RM are analogous to the "semantics" of LEGO bricks. The entire set of all possible combinations of a set of bricks is (even if most of these combinations are meaningless) a significant number. But a few combinations like the construction of a car is in a specific domain a valid and meaningful plan. And these plans are the LEGO versions of archetypes [Bea02, Gar].

The following figure 2.5 visualises the relationship between the RM and the AM.

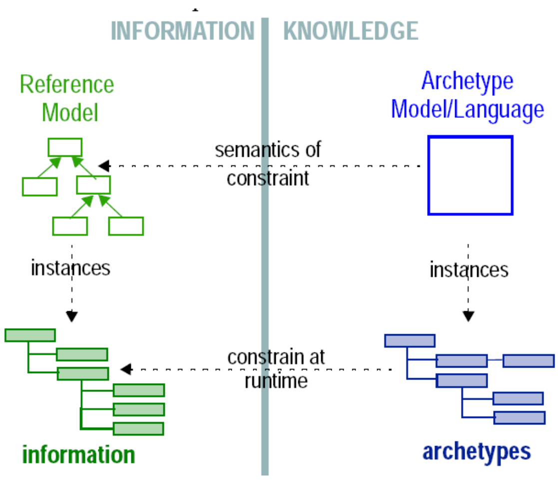

Figure 2.5: The two levels of the openEHR approach.

The "ADL and the AOM are open specifications of openEHR. They have been adopted by CEN TC/251, the European standards agency Health Telematics Committee for use in its revised EN 13606 Electronic Health Record standard. They are now in the process of being standardised by the ISO" [oF06].

3. **The (TOM)**, as described, above is included in the AM. Templates are logical and locally designed models of user forms - and are described in terms of choices of archetypes with associated data captured on a particular form. Usually, the template design is linked to the design of corresponding screen forms [BH07b].
   According to [BH07b] templates include the following semantics:

   - archetype 'chaining': choice of archetypes to make up a larger structure, specified via indicating identifiers of archetypes to fill slots in higher-level archetypes;
   - local optionality: narrowing of some or all 0..1 constraints to either 1..1 (mandatory) or 0..0 (removal) according to local needs;
   - tightened constraints: tightening of other constraints, including cardinality, value ranges, terminology value sets, and so on;
   - default values: choice of default values for use in templated structure at runtime.

   At runtime, templates are used with archetypes to create data and to control its modification.

4. **The EHR Query Language (EQL)** is a query language for EHRs. Listing 2.1 provides an example of the EQL and it is introduced here according to [MFBH07]. The OpenEHR specifications have been developed to standardise the representation of an international EHR. The language used for querying EHR data is not yet part of the specification. Therefore, Ocean Informatics developed a query language known as EHR Query Language (EQL) to fill this gab, it is a declarative language supporting queries on EHR data. EQL is neutral to EHR systems, programming languages, and system environments and it depends only on the openEHR archetype model and semantics. The EQL can be used for any archetype based concept. The use of a common RM, archetypes, and a companion query language, such as EQL improves the semantic interoperability of EHR information.
   EQL is for expressing the queries used for searching and retrieving the clinical data found in archetype based EHRs. It is applied to the openEHR Reference Model (RM) and the openEHR clinical archetypes. The EQL is designed as a common language used for expressing clinical data requests across multiple openEHR-based applications.
   The syntax is influenced by Structured Query Language (SQL), XML Query Language (XQuery), and Object Query Language (OQL). An

example of an EQL query is given in the listing 2.1 below.

```
SELECT COUNT(e/ehr_id)
FROM EHR e
CONTAINS (COMPOSITION c[openEHR-EHR-COMPOSITION.problem_list.v1]
CONTAINS EVALUATION e[openEHR-EHR-EVALUATION.problemdiagnosis.v1]
AND
COMPOSITION c1[openEHR-EHR-COMPOSITION.report.v1]
CONTAINS OBSERVATION o[openEHR-EHR-OBSERVATION.laboratoryhba1c.v1])
WHERE
e/data/items[at0002.1]/value/value=diabetes mellitus
AND
c1/context/other_context/items[at0006]/items[at0013]/value > current-date()
     -P1Y
AND
o/data/events[at0002]/data/items[at0013.1]/value/numerator > 7
```

Listing 2.1: EHR Query Language example.

2.2.3 Interoperability and other Standards

The openEHR architecture for electronic health records is not a standard itself. It is based on the result of many years of research from national and international "standardization" organizations and influences the work of these. It is similar to Linux or Apache [Bea07a].
The major outcome of many years of research (figure 2.2) was that interoperability is an important factor for the success of EHRs. This position became even more determined through an economic analyse [WPJ+05].
Interoperability, the support sharing and exchange of EHR information within EHRs (and not only between one particular implemented system and another) includes also the [Inf06]:

- Sharing of the patient's health information between health professionals in a multi-disciplinary shared-care environment.
- Interoperability between organisations within an enterprise, a regional or national health system.
- Support of interoperability between software from different providers.

Therefore, the main focus of EHR standardisation is on interoperability [Sch04]. Some of these standards do already exist, others are currently under development by national and international Standards Development Organisations like the International Organization for Standardization (ISO), European Committee for Standardization (CEN), and HL7. The openEHR model and specifications provide these types of interoperability through functional interoperability (the transmitted health information, readable by humans) [Inf06]. This is also possible through the use of HL7 v2.x messaging

[Sch04] and semantic interoperability (computers understand and automatically process transmitted health information). There are other approaches like HL7 version 3, OMG's CORBA 3 and CEN EN 13606 "Electronic Health Record Communication" which are architectural approaches for semantic interoperability. However, semantic interoperability is essential to enable local processing of the shared data [Inf06]; this is a pre-requisite for intelligent decision support and care planning.
In contrast, at present, almost all EHRs are based on proprietary information models within EHR systems, with almost no interoperability between EHR systems and almost no ability to share EHR information beyond the immediate boundary of a single organisation. It is often impossible to share EHR information between different disciplines within a single organization or between different applications within a single clinical information system [Sch04, Les]. Therefore, interoperability on both levels is essential to make information in its entire breadth shareable and useable.
In the following, the standards for EHR and related technologies are discussed and it is mentioned how they influence(d) the openEHR architecture and vice versa.
Figure 2.6 illustrates the relation between different standards.

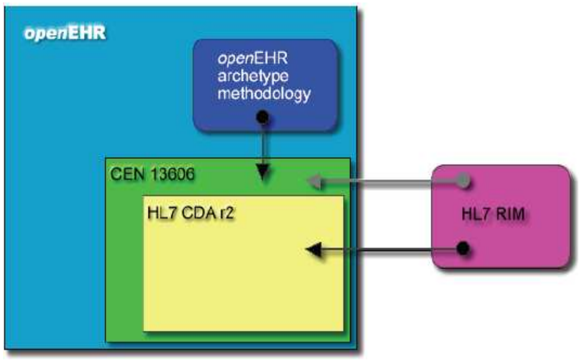

Figure 2.6: openEHR and other standards. Source: [BV07].

2.2.3.1 CEN

The European Committee for Standardization (CEN) is a non-profit technical organization and it has a membership of 30 European countries[4]. It has published two generations of EHR standards (1995 and 1999) [oF06].
In 2002, CEN made the decision to revise the 13606 pre-standard (specification for exchange of EHR Extracts, not a specification of a full EHR system) and upgrade it to a full normative European standard. For the openEHR Foundation the most important aspect of the revision project was a decision

[4]http://www.cen.eu/cenorm/members/members/index.asp

to adopt the openEHR two-level modelling, known as the 'archetype methodology', and also to incorporate some of the openEHR Reference Model into the revised CEN standard [neh06, Sch04]. But for an entire EHR systems there are more requirements to define, this includes version management, workflow management, and interfaces to other systems. For example, the use between openEHR systems does not yet cater properly for version information. The combination of CEN, openEHR, and HL7 has led to an intention to harmonise the proposed new standard with both openEHR (reference model and archetype approach) and with HL7 (mainly the Clinical Document Architecture). ADL and the AOM are open specifications of openEHR. They have been adopted by CEN TC/251, the European standards agency Health Telematics Committee for use in its revised EN 13606 Electronic Health Record standard. They are now in the process of being standardised by ISO [Sch04, Inf06].

2.2.3.2 Health Level 7 (HL7)

HL7 is an American standardization organisation which is accredited by the American National Standards Institute (ANSI). Its main focus is to develop specifications for application-level messaging between health information systems. These specifications are also applied for other areas such as clinical documents and decision support. The versions 2.x (an xml schema) of its messaging standards are in wide use in the USA and around the world. Its typical use is between information systems inside the same hospital, and between hospitals and external laboratories [oF06].
In 1997, they started to develop new HL7 CDA standards. The aims of these new standards are still the messaging between applications but it is based on formal models, including a reference information model (RIM). The HL7 CDA defines the structure and semantics of a clinical document (such as a discharge summary, progress note, etc.) for the purpose of exchange. It is similar to the "Composition" class in CEN 13606 and openEHR. A CDA document can include text, images, sounds, and other multimedia contents. It can exist independently outside the transferring message and can also be transferred within a message [DAB+06]. There are two releases so far, R1 and R2. But it does not address significant requirements in this area, such as distributed version control, flexible EHR Extract structures, archetypes, or querying [oF06].

There is a current (2006/7) effort within ISO to produce a harmonised Data Types specification for HL7, CEN and openEHR [oF06].

2.3 Ocean Informatics openEHR Work Suite

The Ocean Informatics openEHR work suite, represented in figure 2.7, is a view of the entire EHR system with its components. Below, all used components are described, based on (`http://www.oceaninformatics.com`) and its usage in this study.

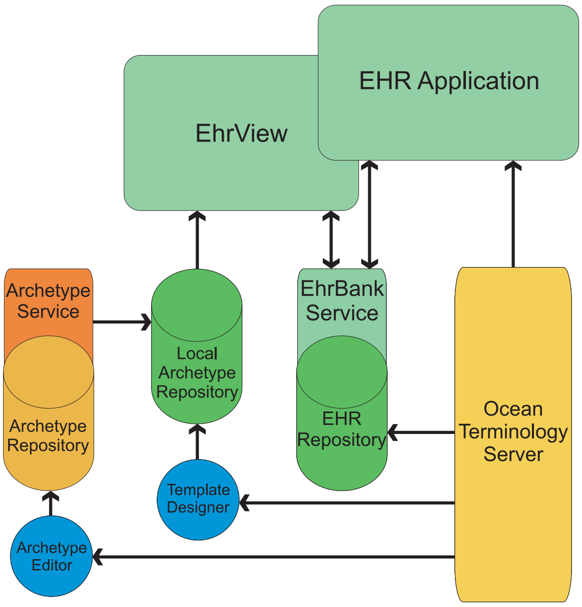

Figure 2.7: Architecture of the openEHR approach. Source: `http://www.oceaninformatics.com`.

1. The EHRbank is a central part of the EHR - system, it is for storing health records and utilising the openEHR framework (release 1.0.1). The EHRbank manages versioning, accountability, privacy and security - and provides an electronic health record service to any applications that are archetype enabled. The EHRbank interacts with other components to manage security, terminology access, demographics access, EHR querying, and interoperation with third party EHR services.

2. Ocean Terminology Server Ocean Informatics has developed a terminology service which can be accessed from any application. The terminology server restricts the set of possible terms used to populate a given data field based on the constraints expressed in archetypes - while allowing browsing of the terminology within this 'knowledge environment'. Ocean Terminology Server consists of 3 components:

    (a) A terminology server

    (b) A termset query editor

(c) An application widget which can be used by clinical applications

3. Archetype Repository Ocean Informatics and Central Queensland University have developed an ontology based archetype repository including an archetype finder which provides the finding of archetypes and ensuring their semantic correctness, and presents them in different formats, such as adl or xml, and in different views similar to the html view or as a mind map. This service, available already in a number of languages, is based on a protégé ontology, which automatically generates the interface.

4. Template Designer. As previously described, templates are an important link between a clinical service and the openEHR approach. Ocean Informatics designed a modelling tool for templates. The main use for the tool in this study is to develop templates, the associated (generic) graphical user interfaces (GUI), and to extract, using this tool, the template data schema (TDS) for each developed template. The development of templates works via drag and drop, it provides the functionality to adapt each archetype within the template according to the usage. In other words, it allows to set several data items to zero.
   The GUI source code is available in the programming languages vb.net and c♯. However the most interesting functionality is the export of templates to template data schemas. Based on the TDS files, it is possible to produce XML instances which can be stored in the EHRbank. A brief overview of Extensible Markup Language (XML) Technologies is given in the next section.
   The TDS files developed in this study can be in the appendix A.1 and the conversion from information to XML files based on the TDS can be found in chapter 4.

5. The Archetype Editor is a multi-lingual tool for expert clinicians to author clinical data specifications and produce archetypes. It can also be used for specialisations of archetypes. It provides different views of archetypes like as HTML, XML and OWL as well as save as ADL and XML.

2.4 Extensible Markup Language (XML) Technologies

This section introduces the Extensible Markup Language (XML) with the XML Schema, and the XML Path Language (Xpath). As described above, the Template designer is able to produce a Template Data Schema (XML Schema file) from templates. This schema is used to produce XML files for the transfer of data from different sources to the EHRbank.

2.4.1 XML

The "Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere"[5]. XML is a free, platform independent, open standard [BYC+] provided by the World Wide Web Consortium (W3C). It addresses the mark-up of data by adding structural information to it [NÖ7]. XML is extensible because it does not provide a set of markup tags as HTML [JRH] or LATEX.
To illustrate the use of XML, an example is given below. The listing 2.3 below shows an XML file of a simple mail. It consists of a header and a body which are described in more detail in the next subsection .

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <email>
3   <header>
4     <To>John-Q@public.com</To>
5     <CarbonCopy>Max@mustermann.de</CarbonCopy>
6     <Subject>Simple Email</Subject>
7   </header>
8   <body>This is the body of a simple email</body>
9 </email>
```

Listing 2.2: XML file example for showing the structure.

To make it valid for the XML schema, which is introduced next, the second line needs to be replaced with this one:

```
<email xsi:noNamespaceSchemaLocation="simplemail.xsd" xmlns:xsi="http://www
    .w3.org/2001/XMLSchema-instance" >
```

Listing 2.3: xml file suitable with the given xml schema.

2.4.2 XML Schema

The XML Schemas express shared vocabularies and allow machines to carry out rules made by humans. They provide a means for defining the structure, content and semantics of XML documents [FW]. The XML Schema language is sometimes referred as XML Schema Definition (XSD). XML Schemas itself are written in XML. The purpose of an XML Schema is to define the legal building blocks of an XML document.
An XML Schema defines[6]:

[5] www.w3.org/XML/
[6] http://www.w3schools.com/schema/

- elements that can appear in a document
- attributes that can appear in a document
- which elements are child elements
- the order of child elements
- the number of child elements
- whether an element is empty or can include text
- data types for elements and attributes
- default and fixed values for elements and attributes

Figure 2.8 shows the xml schema as a map and visualizes the concept. The listing 2.4 represents the same, it shows the XSD of the previously introduced XML file.

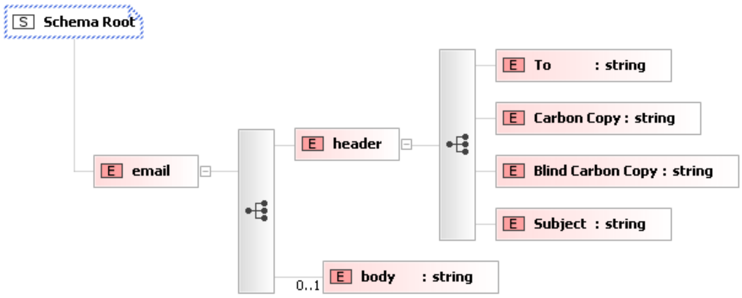

Figure 2.8: XML schema sample as map. Built with Liquid XML Editor.

```
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/
    XMLSchema">
  <xs:element name="email">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="header">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="To" type="xs:string" />
              <xs:element name="CarbonCopy" type="xs:string" />
              <xs:element name="BlindCarbonCopy" type="xs:string" />
              <xs:element name="Subject" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element minOccurs="0" name="body" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Listing 2.4: XML Schema file example for listing 2.3.

2.4.3 XML Path Language

The main purpose of the *XML Path Language* (XPath) [DC99, CBF+] is to navigate through the tree structure of an XML document, to address certain nodes, and to select parts of the document using a simple declarative syntax. For example, the path expression `//header` would evaluate to a sequence containing all `header` elements (including their child elements) from the XML document shown in listing 2.3.
As the name XPath already indicates, path expressions are elementary constructs of the language. A path expression can be used to locate nodes within XML trees and consists of a series of one or more steps, separated by `/` or `//`, and optionally beginning with `/` or `//`. Steps are defined as axis steps or filter expressions. Axis steps define the direction of movement in the tree and filter expressions test if a selected node holds a specified condition. An axis step returns all nodes that are reachable from the context node via a specified axis and that fulfil the filter expression [HM04].

This chapter provides an overview of the openEHR approach, tools which are used in this thesis, and related technologies. Additionally it is shown where the advantages of this approach lie - interoperability on semantic and functional level.
The next three chapters show how an openEHR-based EHR system could be introduced to a clinical service. The next chapter provides a description of an emergency department regarding to its produced data, patient flow, and systems. It also shows the connection to the openEHR archetypes.

Chapter 3

Analysis of a Clinical Service and corresponding openEHR Archetypes

In the first section of this chapter, the study analyses the patient flow, the paper-based and electronic records, and the systems and corresponding processes in the Emergency Department (ED). For the analysis, of the ED several meetings were conducted with a former nurse of the ED, now clinical project officer at the IT Department. In these meetings, the ED has been viewed for several times and introductions of the systems and forms were given. In addition to this, the governmental website: `http://www.health.vic.gov.au/yourhospitals/emergency/index.htm` for ED regulations has also been surveyed. The study investigates how the data items are matched from different sources in a clinical service to existing openEHR archetypes (figure 3.1).

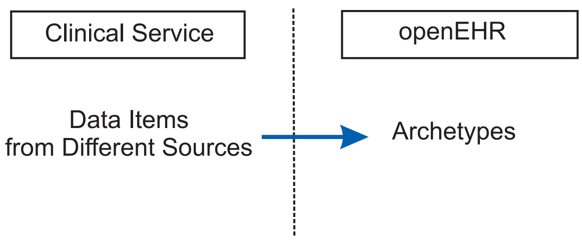

Figure 3.1: Link between clinical services and openEHR archetypes.

There are a few different systems used in the ED. Three of them will be discussed in this section.

In the second section, the study examines the different approaches for the categories of usage of the openEHR EHR system and the management of data in these different cases and additionally gives an example of usage with the regarding components.

3.1 Emergency Department

The ED was chosen as the setting because of its versatile systems, processes, and gathered data. The ED covers many aspects which are also useful for other departments such as the monitoring of vital signs.

3.1.1 Patient Flow, Electronic and Paper - Based Records

With regard to the description of the patient flow in Australian emergency departments[1] and the ED in Austin Health, figure 3.2 represents the patient flow with the corresponding systems and forms.

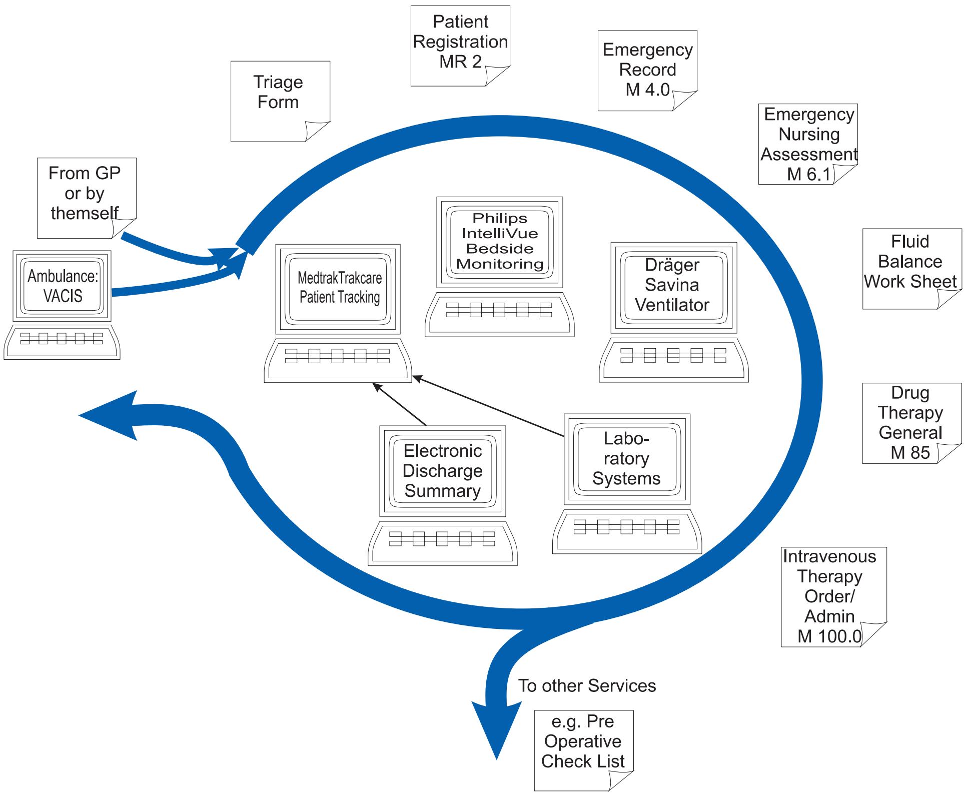

Figure 3.2: Patient-flow in the emergency department.

According to [GoHSA08], public hospital emergency departments meet the immediate health care needs of the community. This includes the treatment of medical emergencies and less urgent cases if alternative care is not available.
In the Austin Health ED, all patients who arrive the ED are assessed on arrival by a nurse. The nurse identifies the urgency of each patient's health

[1] http://www.health.vic.gov.au/yourhospitals/emergency/index.htm

condition and classifies the case into a triage category. The categories are from 1 to 5, 1 stands for resuscitation (patient unconscious e.g. heart not beating, barely breathing, ...) and must be seen immediately. In all other cases the patient must wait in the treatment waiting area. This classification and some other registration data such as name, sex, and language are entered into the patient tracking system Medtrak (section 3.3) and printed on the triage form afterwards.
Depending on the triage category, the patient proceeds to the nursing assessment (also known as cubicle assessment). The nurse observes the patient's social history, skin, renal, respiration, Gastrointestinal Tract, Cardiovascular System, Central Nervous System, and High Osmotic Pressure Chromatography there. The results are entered into the triage form. Some patients may also have blood tests, urine analysis, x-rays, and/or other investigations. There are systems which are connected to Medtrak for example Agfa pacs web 1000 and different laboratory systems. This does not receive further consideration at this point because they already interact on the functional level.
After the nursing assessment a medical doctor also assesses the patient regarding to the observations made by the nurse. These findings are noted in the Emergency Record form M 4.0 (figure 3.2). Some patients are seen by other health professionals like physiotherapists or social workers.
Afterwards the monitoring of the patient conditions begins and monitoring notes/comments are entered into Emergency Nursing Assessment form M 6.1. The Philips bedside system provides the patient's vital signs data in combination with Dräger Savina ventilation system. These systems are described in section 3.1.3 in more detail.
According to the assessment conducted by the medical doctor, the drug therapy, the fluid balance examination, and/or the intravenous therapy would begin also. After this treatment some patients are treated and able to go home, while some are admitted to hospital for further observations.

3.1.2 Data Analysis

After the analysis of the documented and produced data, a spreadsheet was produced (table 3.1). It illustrates the link between the data from different sources within a bigger context like the information [Coi03] for the monitoring process.
This view of the data items allows a few conclusions:

1. Data items in one row may be duplications, if they are measured at the same time. This is not a problem, it shows the retyped data.

2. It shows how data items are distributed for only one measurement.

3. It visualises the connections between electronic and paper - based data.

Bedsite Monitor	*Medtrak*	*Triage*	*Emergency Rec M 4.0*	*Nursing A M 6.1*
BP	BP	BP	BP	Blood Pressure
HR	Pulse	PR	P	Pulse
Resp	Resp	RR	Resp	Respiration
Temp	Temp	Temp	Temp	Temperature
O2 Sat	O2 Sat	SPO2%	SpO2:	SPO2 / O2
	PS	PS		Pain Score
				Patient Condition

Table 3.1: Data items from different sources for the monitoring process.

In this case the gaze is on the monitoring of the patient's vital signs in a very generic case. The order is different in some cases depending on the condition of the patient. All in table 3.1 given data items are produced and/or documented in this process. The blood pressure, pulse, respiration, temperature, and O_2 saturation are the measured vital signs. These are entered into Medtrak at the beginning of the documentation. From there these are printed on the triage form and measured again. Afterwards the medical doctor observes the patient and repeats the measurement. If the patient will stay for a longer period of time he/she will get "connected" to the beside monitoring system where the data is measured and written down on the paper based form: "Emergency Nursing Assessment - cont M 6.1" in certain periods of time. If the patient needs the ventilation system the nurse will write down the settings of the system always if she changes it.
This does not describe the quality of the treatment, it does however illustrate the lack of interoperability of systems and the "media disruption"[2] and how it affect the work of the clinicians. In this general case, there are three media disruptions and some information which is not documented. Therefore, this not documented information is lost.
The analysed systems found in table 3.2 produce or consist of more data than in this case needed. It is because this case study concentrates only on the profile of the introduction not the entire introduction. For example the patient tracking system has a few screens but the triage screen is in the scope of interesting. The bedside monitoring system from Philips produces a few values depending on the settings. In this case only four are considered useful. The ventilation system from Dräger provides 39 setting data items and a few text messages. To summarize the analysis of the data (found in the appendix) the most important facts are as follows (table 3.2).

The paper - based forms (found in figure 3.2) are listed in table 3.3 with its corresponding count of data items.

[2]Media disruption is often referred as retyping and means a change from an electronic to a paper-based form or the other way round.

Electronic-based Forms	*Data Items*
Patient Tracking Systems (Medtrak) (only the triage screen)	36
Bedsite monitor (Philips Intellivue)	4
Ventilator (Dräger Savina)	39

Table 3.2: Brief facts of the electronic-based systems.

Paper-based Forms	*Data Items*
Triage Form	132
Emergency Record	21
Emergency Nursing Assessment	11
Fluid Balance Work Sheet	13
Pre Operative Check List	25
Drug Therapy – General	55
Intravenous Therapy – Order / Admission M 100	9
In Patient Registration MR 2	32

Table 3.3: Brief facts of the paper-based forms.

With this collected data items it is possible to match them to archetypes. This is shown in the appendix A.1.

3.1.3 Systems

This study examines three different systems in the ED. The first one is the Dräger Savina ventilator followed by the Phillips IntelliVue bedside monitoring system and the patient tracking system Trakcare Medtrak. The main focus is to examine the interfaces in order to be able to extract the needed data. But one can also differentiate between push and pull systems. A push system is on hand if a system produces data itself and pushes the data somewhere e.g. a ventilation system. Therefore, it can push the data into the EHR system in certain periods of time. Push data is also present if data is entered directly into the sink for example if one enters data into an openEHR template form. It saves the data archetype based in an EHRbank. A pull system is available e.g. if data is manually entered into a system and the EHR system collects the data itself from the persistent storages of these systems.

3.1.3.1 Dräger Savina Ventilator

The ventilator is a device often used in emergency care. Together with the Philips bedside monitor it provides the monitoring of the vital signs of the patient care and is a push system. The Savina ventilator provides alarm and

text messages, device settings, and real-time data. During the monitoring process, which is described in chapter 4, the data is manually collected by the nurses. Every time when they change the settings at the ventilation system it is written down on the Emergency Nursing Assessment form M 6.1 into the field 'patient condition' like "SIMV 16 x 400 with PEEP 10cm H_2O FIO$_2$ 100%".
The Savina system is currently not connected to any other electronic service that means the handwritten documentation is the only source where the information is retained.
Most of the medical devices from Dräger provide a serial RS 232 C interface. It is called the Medibus and the data stream is ASCII based [KGa05]. The transfer of data is unidirectional, meaning one can not write via the Medibus interface to the ventilation system. However, it is possible to make requests. The interface specification looks as followed [KGa05]:

Port Configuration	
Baudrate	1200, 2400, 4800, 9600, 19200 Baud
Databits	8
Startbits	1
Stopbits	1, 2
Parity	no, odd, even

Table 3.4: Port configuration of the Dräger Savina ventilation system.

The Savina ventilation system provides data through the Medibus configuration which match the manually written documentation. The written information "SIMV 16 x 400 with PEEP 10cm H_2O FIO2 100%" consist of the mode (SIMV, IPPV, CPPV, ...), the frequency in 1/min, tidal volume in liter (x.xxx), PEEP in mbar which is equal to cm H_2O, and Insp. O$_2$ in %.

3.1.3.2 Philips IntelliVue bedsite Monitor

This system is the bedside monitoring system in the ED. It records the vital signs of the patient and presents it on a screen at the patient's bed. This monitoring system is also used in other departments such as ICU. Austin Health is currently working with an external company to develop an interface which converts the measured data into HL7 and XML messages. This module is located at the "Persistent Data Storage" hence this is a pull system. Therefore, and because of the complexity of the data, it is beyond the scope of this thesis and remains open to further research.

3.1.3.3 Medtrak Trakhealth Patient Tracking System

Medtrak as a patient tracking system provides functionalities like the registration of patients and the label printing. For the Emergency Department it supports the management of the patients and interacts with several other applications and systems. For example these are laboratory and image processing systems like pacs. Medtrak provides different screens of the patients for the nurses like registration, ED map, order, triage, and a few other screens for organizing the patient's treatment.
The ED map provides a view of all currently treated and waiting patients with its urgency and action. In this context the more interesting screen is the triage screen. It provides information which are useful for the previously described monitoring process like pulse, temperature, O_2 saturation, and blood pressure. In most cases these are the first measured values and printed on the paper based triage form in the current work flow. Therefore, it is very useful to have these measurements in the EHR system.
On the technical site the Medtrak stores the data in a multi-value object oriented database, cache. Figure 3.3 shows a simplified view of the organization of the systems and how the data flows. To extract some data, which are entered in Medtrak and stored in Caché, SQL views are very useful. These are provided by the IT Department and it gives an easy to use and integrate form of the needed data. Therefore, Medtrak can be seen as a pull system.

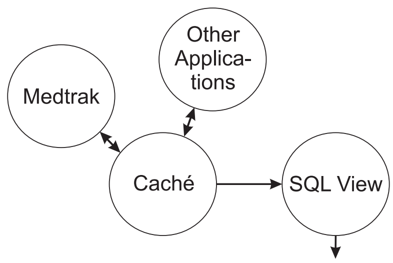

Figure 3.3: Simple view of Medtrak and corresponding systems.

The SQL view of the demographic data which is also entered via Medtrak already exists and is used as an example of extracting data which will be converted later. This SQL-view 3.5 offers such a relational table with the name VARMPatient and a few collums like PatientNumber and PatientSurname.

VARMPatient			
PatientNumber	PatientSurname	PatientGivName1	PatientGivName2
12345678	Public	John	Q
23456789	Max	Mustermann	

Table 3.5: SQL view of the Caché data.

3.1.4 Archetypes

Archetypes are as mentioned a very important part of the two level architecture of the openEHR approach. Therefore, in this section the study describes how the data items match with the corresponding archetypes in the specific field of interest. It also shows where one can find already defined archetypes. The Ocean Informatics openEHR Archetype Finder `http:www.archetypes.com.au/archetypefinder/archetypefinder` and the openEHR archetype Repository `http://svn.openehr.org/knowledge/archetypes`, where all openEHR archetypes are available, provide an easy to use tool for finding archetypes. The archetypes are also available in different formats; as xml and adl which are described in the Foundations. There are also two different views for the Archetypes retrievable as html and as mind map. All these archetypes cover 73 % of all data items which are examined in this study.
The following table shows an example how the matching works. One can see which archetype fits to which data item. This view allows a few conclusions:

1. The matching data items to archetypes.
2. The not existing archetypes.
3. Easy to define templates and specializations.

Nursing Assess M 6.1	*Archetypes*
Blood Pressure Pulse Respiration Temperature SPO2 / O2	openEHR-EHR-SECTION.vital_signs.v1
Pain Score	openEHR-EHR-CLUSTER.symptom-pain.v1
Patient Condition	openEHR-EHR-ITEM_TREE.gas-administration.v1draft

Table 3.6: Data items - archetypes matching table.

The whole matching tables are in the appendix A.1.

To specialise or develop archetypes, the openEHR community offers two prevailing open source archetype editors. One is written using the programming language Java, the LiU Archetype Editor from the Linkoping University, Sweden. The other one is written using the programming language VB.NET and is provided by Ocean Informatics.

3.2 openEHR

Based on the introduction of the architecture of the openEHR approach given in the foundations (2.2.2) and the Ocean Informatics work suite (2.3) there are at least two different categories of usage:

1. The EHR system as a "simple" data sink.
2. The EHR system with a communication server.

3.2.1 Category of Usage and Management of Data

There are different ways to integrate an openEHR system into a service. It depends on several conditions and the aims of the introduction. Two possible solutions are examined below.

- The EHR as a "simple" data sink (figure 3.4).

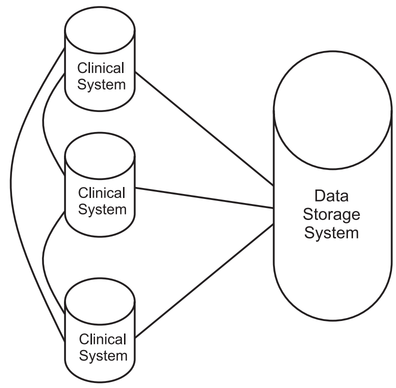

Figure 3.4: n*(n - 1) interfaces in star shaped ordered system.

In this case the EHR system works as a data sink, which means it collects all produced data from different sources like the ventilation system and manually entered data. Sometimes these systems communicate with each other via separate interfaces (figure 3.4). This could lead to n*(n - 1) interfaces and consequently one has to maintain all these. For example if the ED connects all its systems (figure 3.2) star shaped it will maintain 30 interfaces.

  - Management of data in a data sink
    From the management of data point of view there are a few things to discuss. If an EHR system collects data from different systems and duplicates data, problems will occur. Here are some advantages and drawbacks for the duplication of data:
    1. Inconsistency can lead to failures in making decisions for the clinical staff and on the other hand, it is hard to manage the

consistency of data. For example if the interface fails or the delay is too high, one would view old data.

2. This point leads to the need to verify the data,
3. and unexpected changes can lead to rejection of data.
4. Referencing, indexing can lead to trouble if changes occur on one site.
5. It costs massive space to store the data.

On the other hand the duplication of data has some advantages:

1. It can be used as a backup system.
2. The data can be accessed without separate logins.
3. It can provide its own look and feel.

However, to store data in a data sink does not mean that it automatically duplicates the data. For example, the ventilation system produces data, which is not electronically stored and the paper based record could be replaced. It could be a good approach to store it in a data sink.

- EHR with a communication server
To use the openEHR approach with a communication server (figure 3.5) means that it exhausts the possibilities of the currently operating systems with less interfaces. The communication server does not store the data it provides the communication between the systems and transfers the data in the correct format to the EHRbank.

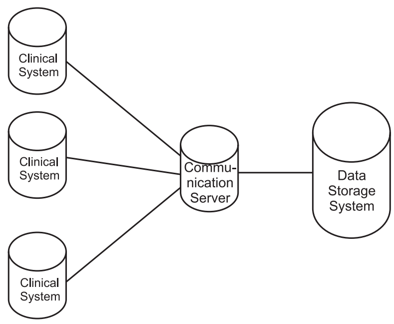

Figure 3.5: n * 2 + 2 interfaces in systems with communication server.

With the previous example of the ED one has to maintain only fourteen. Thus a communication server is suitable from a certain number of systems.

Both uses have assets and drawbacks. In this study the EHR system is a data sink. Therefore, the following example of usage is used in this study.

3.2.2 Example of Usage

The following figure (3.6) represents how an EHR system based on the openEHR approach could be integrated and how to transfer data from other sources.

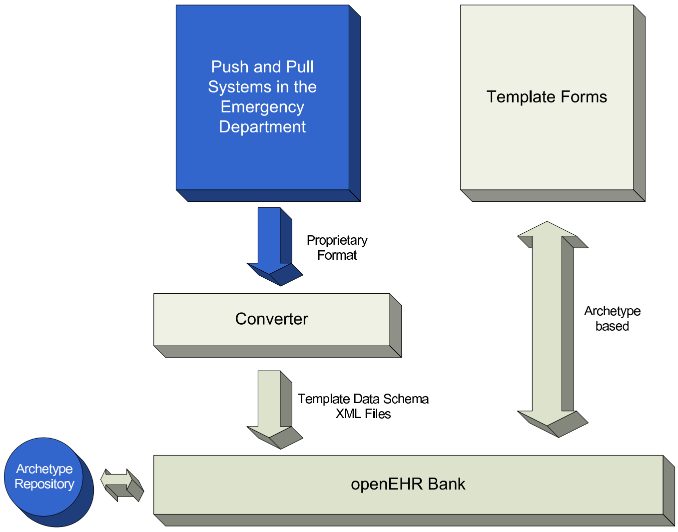

Figure 3.6: Example of an openEHR-based EHR system.

There are different types of systems with different interfaces within a service and the collected or produced data flows to the converter. The converter has the aims to provide interfaces for these systems and parse the received messages to the correct format for the EHRbank. This is the storage of the information and therefore represents the instance of the RM in connection with the archetypes.
From here data can be view via EQL queries.

The next chapter discusses the design of templates and the set-up of an openEHR architecture based EHR system. It also shows the transfer of the extracted data to the EHRbank.

Chapter 4

Design of openEHR Templates and Set-Up of the EHR System

In the previous chapter, the study examined the link between openEHR archetypes and data items from different sources in a clinical service. In this chapter, the study discusses the connections between data items and generic clinical processes in the ED. It shows how templates are built from archetypes with regard to the patient flow in a clinical service. It also shows how the templates visualise the information used in the processes via a graphical user interface (GUI) (figure 4.1).

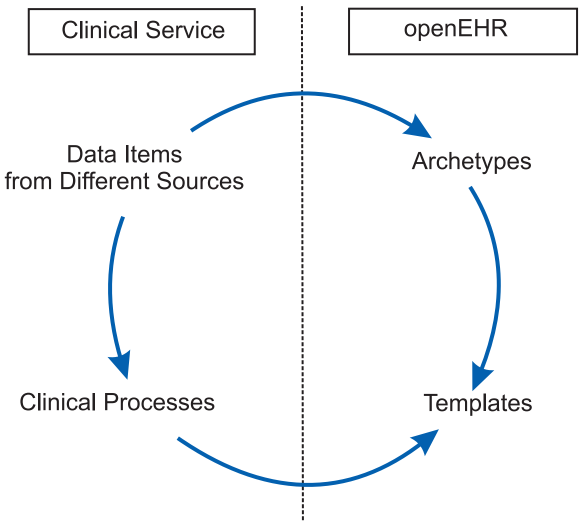

Figure 4.1: Link between a clinical services and the openEHR architecture (archetypes and templates).

Following that, it discusses the set-up of the EHR system including its components. It shows the concrete functionality of the converter (a component of the EHR system) which provides the interfaces to the different systems and a parser for transforming messages for EHRbank. It is followed by the examination of the GUI for the processes and the regarding EQL Queries. This chapter therefore provides the transfer of data from different sources. Furthermore it shows the integration of the EHR system regarding to the different interfaces via a template data schema and the manual entry of data for the processes in the ED according to the patient flow.

4.1 Clinical Processes and Corresponding Data

According to the analysed data items and systems in the previous chapter 3, one can describe clinical processes with its corresponding data items (table: 3.1).

In some cases the data entry presentation is dissimilar to the data presentation. There are clinical processes which have an impact on each other. In other words if a setting on one clinical process like the intravenous therapy changes (e.g. flow rate), this could have an impact on the vital signs of the patient. Therefore, it is expedient to indicate the changes on the vital signs screen. It should be displayed in a very compact and good visible way. This is in harmony with the paper based record "Emergency Nursing Assessment M 6.1" where these changes are recorded in the field "Patient condition". Therefore, one can see that a process consists of one or more data items from push and pull systems and can be influenced by other processes. This is illustrated in the figure 4.2.

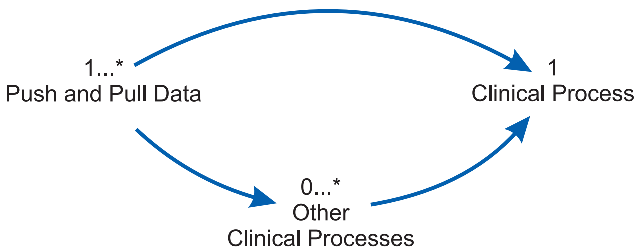

Figure 4.2: Processes with the corresponding data items.

In the following there are descriptions of a few processes with the corresponding data items, systems, and processes displayed in spreadsheets.

The Monitoring of the patient's vital signs, pain score, and condition, which is found in the Nursing Assessment M 6.1 paper based record, is probably one of the most important documentations in the ED, as well as in other departments. The collected data in this process is from push and pull systems and it is influenced by other clinical processes. Therefore, it is qualified as

an example to discuss at this point.
Below the needed data items from different sources and processes are shown:

Type	System/ Form	Data Item
Pull data	Trakcare Medtrak	Blood Pressure, Pulse Rate, Respiration, Temperature, O_2 Saturation, Pain Score
	Philips Bedside Monitoring System	Blood pressure, Pulse, Respiration, Temperature, O_2 Saturation
Push data	Dräger Savina Ventilator	Insp.O_2 in %, Insp. Tidal. Volume in L, Freqeuncy in 1/min, PEEP in mbar, Mode
	Manual Entry (ME)	Pain Score
Other Processes (Push Data)	Drug Therapy ME	Medication, Dose
	Intravenous Therapy ME	Type of Fluid, Additives including Dose, Rate

Table 4.1: Dependencies; data items and processes (Monitoring).

To complete this process two other processes are required which are merely a conversion from the paper based forms to electronic forms.

The Drug Therapy is the clinical process where the nurse administers the drugs which have been prescribed by a doctor.

Type	System/ Form	Data Item
Pull	-	
Push	Manual Entry	DOB, Sex, Patient Weight, Height, Date, Route, Medication, Dose, Frequency, Drug Level, Time Level Taken, Dose, Time to be given
Other Process	-	

Table 4.2: Dependencies; data items and processes (Drug Therapy).

The approach of administering fluids directly into the patient's vein is called intravenous (IV) therapy. The rapid passage of the medicine through the veins gives it an advantage over the much slower processes of the stomach and some fluids cannot be given by eating or drinking[1] [Dep06]. The following table 4.3 shows the recorded data items.

Type	*System/ Form*	*Data Item*
Pull	-	
Push	Manual Entry	Line Number, Flask Number, Type of Fluid, Volume of Flask (ml), Additives including Dose, Rate, Date of Administration, Time Started, and Time Finished.
Other Process	-	

Table 4.3: Dependencies; data items and processes (Intravenous Therapy).

4.2 Templates and Specialized Archetypes

Templates are described as a locally produced constraint specification which articulates which archetypes go together on a screen form or message specification (compare chapter 2).
Archetypes express what can be maximally documented about a topic. The scope of the topic must be completely clear [oF]. However, in some cases especially in the research area more than the given data items are required and therefore a specialization or extension of existing archetypes is needed in these cases.
Figure 4.3 presents the emergency department monitoring template as described above as a mind map. It shows how the archetypes are linked within the monitoring template for the ventilation data. This example is also used for the later discussed use of the template data schema.
Because of the principle of the maximum dataset some of optional items of the archetypes are "switched off" in the template. Only the items which are useful in this context are shown. The Ocean Informatics Template Designer offers a few different views of templates. An html view of the developed template is given in the appendix.

[1] A brief description of the intravenous therapy can be found here: `http://www.time.com/time/magazine/article/0,9171,736319,00.html`

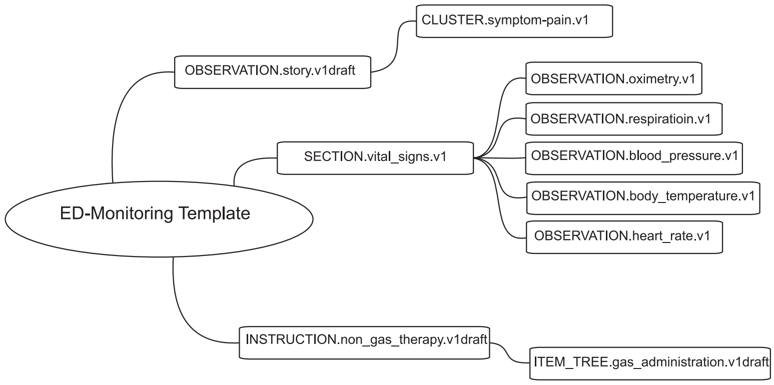

Figure 4.3: The ED monitoring template as mind map.

The basis of the template in figure 4.3 is an encounter composition archetype. This archetype provides unrestricted slots, this example shows the adding of the non drug therapy instruction and two other archetypes.

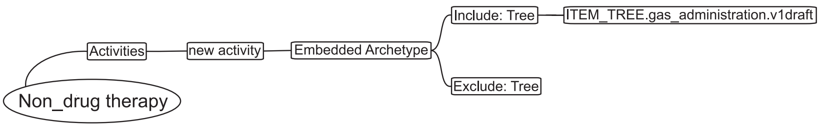

Figure 4.4: The non-drug therapy archetype as mind map.

The Non-drug therapy archetype (figure 4.4) is an instruction archetype. Instruction archetypes may consist of a protocol and one or more activities. These activities can be described directly or through embedding of other archetypes. In this case the gas administration archetype is included.

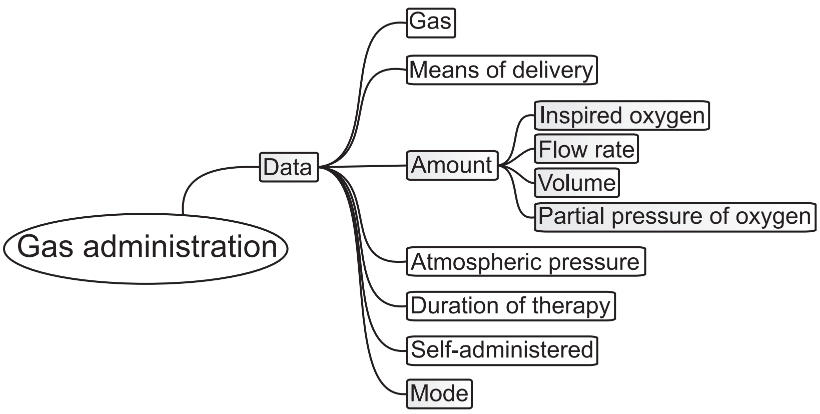

Figure 4.5: The gas administration archetype as mind map.

The gas administration is a tree structure archetype (figure 4.5). It describes the general administration of gas e.g. oxygen or nitrous oxide. For the data which is going to be stored, the cluster ”Amount”, which describes the inspired oxygen, the flow rate, and the pressure, is required. The volume is missing and has to be added to the ”Amount” as well as the item ”Mode”.

Based on these developed templates there are a few possibilities to make them available on a screen and for messaging. These are described in the following sections.

4.3 Templates and Graphical User Interfaces (GUIs)

The openEHR templates may represent clinical processes through combining and constraining archetypes. These templates can be used for messaging and for visualising.

4.3.1 Approaches for GUIs and Applications

Depending on the category of usage it might be useful to display the data on a GUI for user interaction e.g. for data entry. Therefore, the openEHR Community and Foundation are interested to standardise the data entry and data representation. Through this the implementation of GUIs should be far more straightforward and there are international approaches underway[2].
In general, the template form GUI should provide a minimum of the same functionality as the paper based form. To utilise the full potential of interoperable EHR systems they have to be accepted by their users, the health care providers. GUIs that support customisation and data validation play a decisive role for user acceptance and data quality [SGHB06].
The GUI for the processes intravenous therapy and drug therapy for example should provide fields for the data items which could be manually entered. However, when a nurse starts a particular therapy or changes the dose, this could affect the vital signs of the patient as shown in 4.1 but this remains open for further research[3].
There are a few different approaches regarding the presentation of GUIs in development. The described approach for a GUI above is completely programming language and technology independent. Currently there are discussions within the openEHR community about web browser based approaches like the Yahoo User Interface (YUI). The YUI provides functionalities which are useful in these cases as well as XForms.
XForms are the next generation of web forms [BI07]. They split the traditional XHTML forms into three parts - the XForm model, instance data, and

[2] `http://www.openehr.org/wiki/display/dev/User+Interface+and+openEHR+data`

[3] An example how a EHR GUI could look like is given in the diplom thesis from Jasmin Buck ref

user interface. Therefore, they transmit XML files which can be based on the developed template data schemas. Additionally, XForms are platforms and device independent, separate data, and logic from presentation[4]. Figure 4.6 illustrates this model.

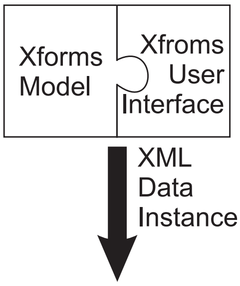

Figure 4.6: The XForms model.

4.3.2 Template Data Schema for Messaging

For messaging purposes it is possible to create a template data schema which is an xml schema file based on the template. This is very important for the previously mentioned XForms as well as for the transfer of data from different sources. The data can be written into xml files which are based on the xml schema. These files contain the actual data and are ready to be sent to the EHRbank.

4.4 EHR System

In the analysis part of this thesis (chapter 3) an example of usage is presented. Based on this and on the previous sections an example will now be viewed in more depth. It provides an overview of the entire system.

4.4.1 Overview

This figure 4.7 shows a detailed view of the EHR data flow with the interfaces to the already existing Systems in ED, the messages, and the processing of data.

[4] hppt://www.w3schools.com/xforms/xforms_intro.asp

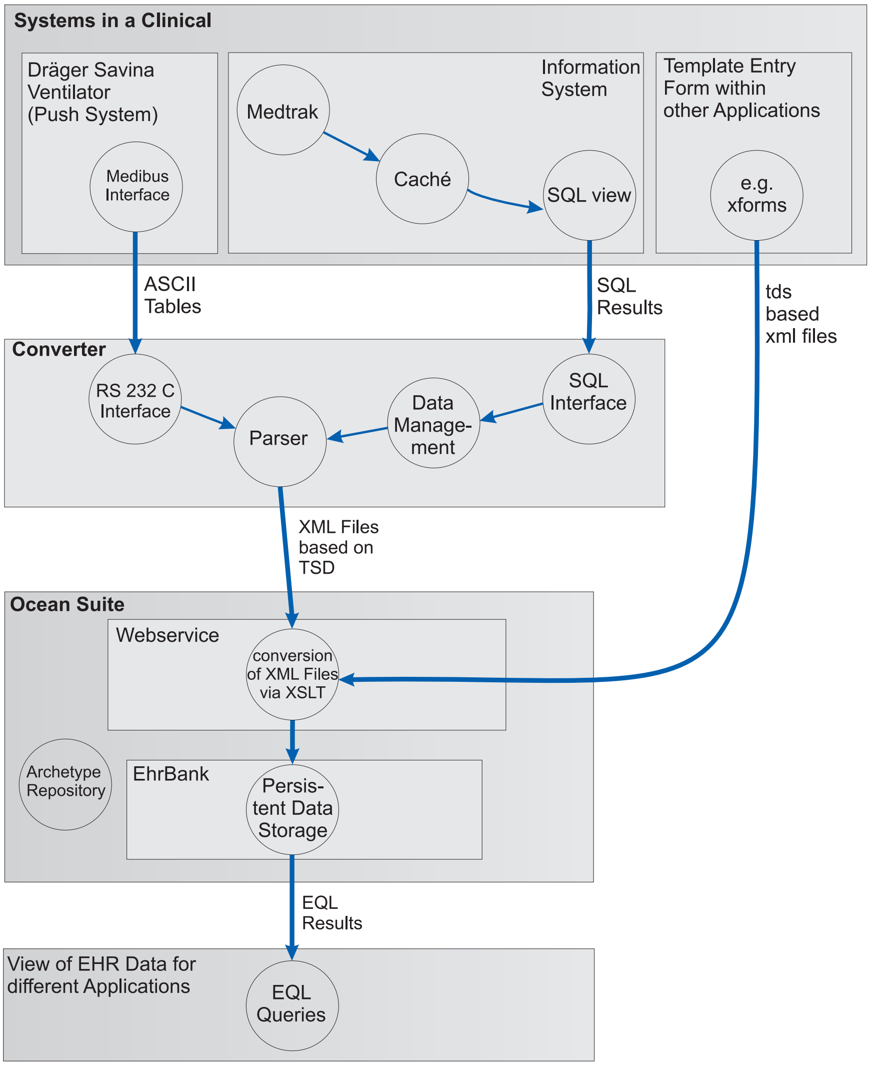

Figure 4.7: EHR architecture overview.

4.4.2 Systems within a Clinical Service

A clinical service like the ED has many different systems which may be categorised into push and pull systems. In figure 4.7 there is one pull system (patient tracking system) and a push system (ventilation system). Additionally, there is a template as an XForm for data entry which can be within other browser based applications or by itself. XForms produce xml files for the transfer and therefore the data can be sent directly to the EHRbank.

4.4.3 Converter

The aim of the converter is to receive and manage the data from different systems via proprietary interfaces. It converts the data into template data scheme based XML files and sends them to a webservice which provides a XSLT script for transforming the data for EHRbank. In the following, the conversion from an ASCII table message to a tds based xml files is described.

4.4.3.1 Interfaces and Parser

The Dräger Savina ventilation system provides push data in ASCII tables. The implementation of this interface is given in the next chapter 5. At this point the tds and the resulting xml file with the data from the ventilation system are discussed.

The listing 4.1 shows an extract of the tds of the ED monitoring template 4.3. It provides the needed information for the xml file with the data for the flow rate.

```
<xs:schema xmlns:oe="http://schemas.openehr.org/v1" xmlns="http://schemas.
    oceanehr.com/templates/" elementFormDefault="qualified" targetNamespace
    ="http://schemas.oceanehr.com/templates/" xmlns:xs="http://www.w3.org
    /2001/XMLSchema">
  <xs:import schemaLocation="Structure.xsd" namespace="http://schemas.
      openehr.org/v1" />
  <xs:element name="Ventilation">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Non-drug_therapy">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" name="new_activity">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="name">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element default="new activity" name="value"
                              type="xs:string" />
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="Gas_administration">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element minOccurs="0" name="Amount">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element minOccurs="0" name="Flow_rate">
                                  <xs:complexType>
                                    <xs:sequence>
                                      <xs:element name="name">
                                        <xs:complexType>
                                          <xs:sequence>
                                            <xs:element default="Flow rate"
                                                 name="value" type="
                                                xs:string" />
                                          </xs:sequence>
                                        </xs:complexType>
```

```
                                    </xs:element>
                                    <xs:element name="value" type="
                                        oe:DV_QUANTITY" />
                                  </xs:sequence>
              <!-- Closing Tags -->
  </xs:element>
</xs:schema>
```

Listing 4.1: XML file example for showing the structure.

The xml file can be produced from the tds (listing 4.2). Row 18 expresses the value for the flow rate (16).

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <Ventilation type="oe:COMPOSITION" archetype_node_id="openEHR-EHR-
       COMPOSITION.encounter.v1draft" xsi:schemaLocation="http://schemas.
       oceanehr.com/templates/ ventilation.xsd" xmlns="http://schemas.oceanehr
       .com/templates/" xmlns:oe="http://schemas.openehr.org/v1" xmlns:xsi="
       http://www.w3.org/2001/XMLSchema-instance">
3    <Non-drug_therapy>
4      <new_activity type="oe:ACTIVITY" archetype_node_id="at0001">
5        <Gas_administration type="ITEM_TREE" archetype_node_id="openEHR-EHR-
             ITEM_TREE.gas_administration.v1draft">
6          <name>
7            <value>Gas administration</value>
8          </name>
9          <Amount type="oe:CLUSTER" archetype_node_id="at0004">
10           <name>
11             <value>Amount</value>
12           </name>
13           <Flow_rate type="oe:ELEMENT" archetype_node_id="at0005">
14             <name>
15               <value>Flow rate</value>
16             </name>
17             <value>
18               <oe:units>16</oe:units>
19             </value>
20           </Flow_rate>
21         </Amount>
22       </Gas_administration>
23     </new_activity>
24   </Non-drug_therapy>
25 </Ventilation>
```

Listing 4.2: XML file example for showing the structure.

This xml file can be sent now to the webservice where it will be prepared for EHRbank. A complete tds of the ED monitoring is available in the appendix A.4.

4.4.3.2 Management of Data

The management of data is very important because one has to avoid writing the same data more than once. In chapter 3 the management of data with its advantages and disadvantages was discussed.

The Medtrak data and EHRbank data can become inconsistent and therefore one has to check these data. This can be done through comparing the data

in certain periods of time or event based. Otherwise the integrity of the systems can not be ensured.
On the other side there is the ventilation system which is a push system and therefore continuously produces relevant data for the EHRbank. According to the monitoring process and the field of the patient condition, where the changes of the ventilators settings were documented, not all of them are useful for this example. There are a few possibilities how to organize this.

1. One can store all produced data even if not all of it will be needed for this application. Each application can constrain the dataset for its needs. In other words, these applications can select only the data which is useful in this case via EQL.

2. The not needed data can be discarded.

4.5 Summary of the Analysis and Design

The analysis of the ED shows the gathered data and clinical processes according to the patient flow. There are also push and pull systems identified in a clinical service. Most of these systems have proprietary interfaces. The gathered data items are mapped to openEHR archetypes. Theses identified or extended archetypes were combined and constraint to templates. Templates build the basis for the transfer of the data from different sources via template data schema. These schemas are used for producing the xml files for the EHR data storage (EHRbank). After the data is sent to the EHRbank the data is available via the EQL queries.

The next chapter povides the foundations for a prototype implemtentation based on this work.

Chapter 5

Foundations for Prototype Implementation

This chapter represents the foundations for a prototype implementation of the previously introduced converter. This means it shows significant parts of the implementation. These pieces are written using the programming languages VB.NET and C♯. It shows the functions of the converter with its interfaces and parsers.
The converter provides the connection between different systems and the EHRbank. Figure 4.7 shows that some programs do not need this converter because the messages are already in the correct format. XForms are based on the given tds for data entry for example. Clinical applications can also use the EHRBank as its data storage. This converter could be replaced by software like BizTalk in commercial solutions. The converter consists of an RS232 interface which is introduced in the analysis part (section 3.1.3) and a parser. It is written in VB.NET. The modality for transferring data from other systems to the EHRbank does not vary much, therefore it is enough to show one.
The parser has the task to transform the given messages (ASCII tables, HL7 files, SQL results, . . .) into xml files based on the produced templates data schemas.
It starts with the interface for the Dräger Savina ventilation system and devolves to the conversion of the received data to xml files.

5.1 Interface (RS 232 Medibus)

Interfaces provide the communication between systems. They are the points of contact where a systems meets its environment or where (sub) systems meet each other. Below are excerpts of the implementation of this interface followed by the implementation of the parser which is written in C♯.
The Medibus is a proprietary interface by Dräger for the RS232 port. This

implementation is based on two open source examples[1,2].
The program has to import a few classes (programming language constructs used to group related fields and predefined functions) for the communication with the COM ports. The aim of the class "InteropServices" is to simplify interoperation between .NET Framework components and unmanaged code[3].

The Text namespace contains classes representing ASCII, Unicode, UTF-7, and UTF-8 character encodings[4]. In this case the ASCII representation is required.
The namespace for threads is imported for reading and writing purposes.

```
Imports System.Runtime.InteropServices
Imports System.Text
Imports System.Threading
```

Listing 5.1: Imports for communication via RS233.

This functions overloads the open function to provide the settings for the interface.

```
    Public Overloads Sub Open(ByVal Port As Integer,
        ByVal BaudRate As Integer, ByVal DataBit As Integer,
        ByVal Parity As DataParity, ByVal StopBit As DataStopBit,
        ByVal BufferSize As Integer)

        Me.Port = Port
        Me.BaudRate = BaudRate
        Me.DataBit = DataBit
        Me.Parity = Parity
        Me.StopBit = StopBit
        Me.BufferSize = BufferSize
        Open()
    End Sub
```

Listing 5.2: Sample source code for the RS232 port settings.

Afterwards it is possible to use the given threads for asynchronous reading and writing. For this purpose a few dlls needs to be imported e.g. for the state of the COM port.

The received data can be stored in a file or sent directly to another function for creating the xml files.

[1] http://www.microsoft.com/downloads/details.aspx?FamilyID=075318ca-e4f1-4846-912c-b4ed37a1578b&displaylang=en
[2] http://www.codeworks.it/net/VBNetRs232.htm
[3] http://msdn2.microsoft.com/en-us/library/sd10k43k.aspx
[4] http://msdn2.microsoft.com/en-us/library/system.text.aspx

```
<DllImport("kernel32.dll")> Private Shared Function GetCommState( _
    ByVal hCommDev As Integer, ByRef lpDCB As DCB) As Integer
End Function
```

Listing 5.3: Sample source for dll import.

5.2 Parser: ASCII Tables to XML

As described in the previous chapter, the data structure of the extracted data needs to be transformed into xml files based on the tds, so that the data can be sent to EHRbank after a second transformation via XSLT.
As mentioned the data from the ventilation system comes in ASCII tables. There are different solutions available. One can use XML editors which provide an automatic generation of source code for parsing the xsd files (table 5.1). With this generated code it is possible to produce xml files based on the tds and fill with data.

XML-Editor	*C++*	*C ♯*	*Java*	*vb.net*
Altova XMLSpy	✔	✔	✔	✗
Liquid XML Studio	✔	✔	✔	✔
oXygen	✗	✗	✗	✗

Table 5.1: XML-editors code generation.

The oXygen editor does not provide any of these functionalities. The Liquid studio editor supports also the functionality for vb.net but one can use the c♯ classes in vb.net, too. Therefore, it does not matter which system is used for this. The following excerpt of the parser is based on the generated source code from the Altova XMLSpy.
In listing 5.4 the source code is given for the production of the xml files for the EHRbank.

```
protected static void Example()
{
   ed_monitoringDoc doc = new ed_monitoringDoc();
   doc.SetSchemaLocation("ed_monitoring.xsd");
   MonitoringType root = new MonitoringType(doc, "", "", "
       ventilator_example");

   root.AddVentilation(new VentialtionType());
   root.Ventialtion.AddNon-drug_therapy(new Nondrug_therapyType());
   ///...
   root.Ventilation.Non-drug_therapy.new_activity.Gas_administration.
       Amount.Flow_rate.Addvalue(new SchemaString("16"));
   doc.Save("ed_monitoring.xml", root);
}
```

Listing 5.4: Sample source code for producing xml files.

The produced xml file is prepared now to be sent to EHRbank where it can be stored and accessed via EQL from/for other application.

Chapter 6

Discussion

6.1 Main Results

The aim of this thesis is to address the empty space of information for the introduction of an openEHR architecture based EHR system to a clinical service. This case study enriches the gap of information via presenting a roadmap, and was the method chosen because it best supports the introduction of openEHR in a dedicated environment, using an example of the emergency department and the ventilation system. This example will demonstrate how an openEHR based EHR could be introduced, how systems could be integrated, how the data could be transferred, and how new clinical application could be developed based on the openEHR approach.

The thesis touches many different aspects, different technologies, systems, providers, and departments. The six month project shows only a very small percentage of the work required to introduce openEHR entirely to a department. The complexity needs to be considered for the introduction of the approach or any other as the proprietary interfaces complicate the integration of systems and the migration of data.
The limitation of the given case study is that adequate tests are missing.
Other possible methods certainly exist such as an iterative model. However this model will introduce one system after another. For example, it concentrates first on the ventilation system, integrates it completely, and than proceeds with the next system.

The ED was analysed according to the patient-flow. It is also possible to analyse the department from other points of view like the nurses, doctors or other roles. The analysis based on the patient-flow is an important perspective (the patient's) and touches many aspects like clinical processes, paper and electronic based documentation. The analysis identified and classified push and pull systems. This reduced the amount of the analysed systems. But there may be other aspects which could also have impact and therefore a classification into these two types is not always suitable.

Chapter 3 introduced different categories of usage. The solution with a communication server could be improved, allowing only a few different interfaces, but is probably only possible in a limited scale of systems. Through presenting the two approaches for categories of usage it addresses an important point that other solutions are possible, for example hybrid architectures.

The results of the design of the link between openEHR and the ED are matching tables for the data items with the corresponding openEHR archetypes. Matching tables are an applicable tool for improving the reuse of archetypes. Higher reuse of archetypes leads to higher interoperability. If, for example, data is stored based on the blood pressure archetype (from the openEHR repository) in Germany and in German, it will also be correctly readable and understandable e.g. in England in English by humans and computers.
The description of the relation between openEHR templates and clinical processes, as shown by the example of the monitoring of the vital signs, is visualized in figure 4.1. Additionally, the influences were identified between clinical processes and the graphical user interface (GUI) representation.

An overview of an openEHR architecture based EHR system is given. Based on this overview, an example is provided on how to transfer data from the ventilation system to the EHRbank. The data is transferred via template data schema based xml files.
XForms and other web-browser-based technologies are mentioned. But the security aspect is not discussed in the thesis. The secure transfer of data from the ventilator to the EHRbank is not difficult in most of the cases. In web-based applications the transfer of data needs to be encrypted via e.g. secure sockets layer (SSL).

Consequently, a profile is given for introducing an openEHR architecture based EHR System.

6.2 Implications for Practice and Outlook

The strength of the EHRs, as an increasingly important role for electronic management of health information, are the delivery of safer, more efficient, better quality health care [neh06, PPTK05]. This could be achieved through less media disruption, less duplication of observations, and more accessible information. The open approach for an EHR architecture seems to meet the requirements.
The main advantages of the openEHR approach are the functional and semantical interoperability. The functional interoperability represents the cor-

rect communication between two or more systems. This is also covered by other approaches like the HL7 v2.x. The openEHR approach also offers the semantic interoperability. It is the ability of two or more computer systems to exchange information which can be comprehended unambiguously by both, humans and computers.

The openEHR Foundation offers no concrete implementation of a clinical application, to achieve interoperability. At this point, companies that build their clinical application upon the openEHR architecture, or even implement this architecture, offer the foundation for the introduction of an openEHR-based EHR.
With the results one can see a roadmap, which provides an example how the openEHR approach for an electronic health record architecture could be introduced to a clinical service and how "new" clinical applications could use these.

Another interesting point to make is who will introduce an openEHR architecture based EHR and who benefits the most? This openEHR concept could lead to more interoperability in many departments, hospitals, general practitioners (GPs), and the entire healthcare chain. It is questionable if the patient or GP will save the money. Therefore it is possibly suitable for cross-hospital concepts, research project, and care centres.
Furthermore it is not shown how far interoperability is possible in reality. Is the fourth level[1] of Walker's model achievable?

After 18 years of research the openEHR approach offers an entirely new and open architecture for an EHR. That means that the specifications are freely available, that there are open source projects, and that it is open for new clinical applications. With the disjunction of information and knowledge, a new generation of application and interoperability is introduced. Now it is up to the vendors to use it.
To sum up, with the given roadmap for the introduction of an openEHR architecture based EHR system, the openEHR approach may be an applicable tool for improving the interoperability within health services.

[1] Machine-interpretable data - transmission of structured messages containing standardized and coded data.

Glossary

ANSI	Is the American National Standards Institute. It is a private, non-profit organization that facilitate the development of standards for the information processing industry.
ASCII	The American Standard Code for Information Interchange (ASCII) is an ANSI binary-coding scheme consisting of 128 seven-bit patterns for printable characters and control of equipment function.
C♯	Is an object oriented programming language from Microsoft for the .NET platform.
CEN	Is the Comité Européen de Normalisation. CEN is a non-profit European standardization organization. It contributes to the objectives of the European Union and European Economic Area with voluntary technical standards which promote free trade, interoperability of networks, etc.
e-health	Is a term for healthcare practice which is supported by electronic processes and communication. The ”e” is today often known as enhanced in comparison to the older meaning electronical. The term encompass a range of services that are at the edge of medicine/healthcare and information technology: Electronic Health Records, Telemedicine, Evidence Based Medicine, etc.
GUI	The Graphical User Interface (GUI) is the bridge between the user and the operating system consisting of graphic images or icons. A GUI allows the user to input information and commands into the computer in a graphical method.
ISO	Is a major International Standards Organization (ANSI is US member) that promote the development of standards to facilitate the international exchange of services and goods as well as in areas of intellectual, technological, and scientific activity.

OQL	Object Query Language (OQL) is a query language standard for object-oriented databases modelled.
RS232	Stands for Recommend Standard number 232. The serial ports on most computers use a subset of the RS-232C standard. Theses serial cable interfaces are used for 25 and 9-pin connections. Most new PCs are equipped with male type connectors having only 9 pins.
VB.NET	Visual Basic .NET is the next generation of Microsoft's Visual Basic programming languages. It is an object-oriented programming language and is not backwards-compatible with VB6.
Waterfall Model	The waterfall model is a software development model. It represents software development based on processes and products. Each process ends with a product as output. The new product becomes the input of the next process. It is introduced by Royce in 1970.
XSLT	The Extensible Stylesheet Language Transformations (XSLT) is a standard subset language of XML designed to allow one XML data structure to be transformed into another. For example, XML files can be transformed into HTML pages.

Bibliography

[Bea02] Thomas Beale. Archetypes: Constraint-based Domain Models for Future-proof Information Systems. Deep Thought Informatics Pty, Ltd Mooloolah, Qld, Australia, 2002.

[Bea05] Thomas Beale. The health record - why is it so hard? In *IMIA Yearbook of Medical Informatics 2005: Ubiquitous Health Care Systems*, pages 301–304. Schattauer, 2005.

[Bea07a] Thomas Beale. openEHR a (defacto) standard for the Global Health Records? http://www.phcsg.org/main/AnnConf07%20Presentations/ThomasBealeAnn07.ppt, 2007.

[Bea07b] Thomas Beale. openEHR a primer. http://www.openehr.org/downloads/presentations/openEHR_primer_sep_2007.ppt, 2007.

[BH07a] Thomas Beale and Sam Heard. openEHR Architecture – Architecture Overview. http://www.openEHR.org/svn/specification/TAGS/Release-1.0.1/publishing/architecture/overview.pdf, 2007.

[BH07b] Thomas Beale and Sam Heard. The Template Object Model (TOM). http://www.openehr.org/releases/1.0.1/architecture/am/tom.pdf, 2007.

[BI07] John M. Boyer and IBM. XForms 1.0. W3C, http:www.w3.org/TR/2007/REC-cforms-20071029/, 2007.

[BV07] Ocean Informatics Europe BV. EHR in Europe a new paradigm. http://www.worldofhealthit.org/docs/presentations/44_freriks.pdf, 2007.

[BYC+] Tim Bray, François Yergeau, John Cowan, Eve Maler, Jean Paoli, and Michael Sperberg-McQueen. Extensible Markup Language (XML) 1.1 (Second Edition). World Wide Web Consortium Recommendation. http://www.w3.org/TR/2006/REC-xml11-20060816, 2006.

[CBF+] Don Chamberlin, Anders Berglund, Mary Fernández, Jonathan Robie, Jérôme Siméon, Scott Boag, and Michael Kay. XML

Path Language (XPath) 2.0. World Wide Web Consortium Recommendation, January. http://www.w3.org/TR/2007/REC-xpath20-20070123, 2007.

[CHI05] The good european health record. http://www.chime.ucl.ac.uk/work-areas/ehrs/GEHR/index.htm, 2005.

[Coi03] Enrico Coiera. *Guide to health informatics*. ARNOLD-A member of Hodder Headline Group, London, Great Britain, 2nd edition, 2003.

[DAB+06] Robert H. Dolin, Liora Alschuler, Sandra L. Boyer, Clavin Beebe, Fred M. Behlen, Paul v. Biron, and Amnon Shabo Shvo. Hl7 clinical document architecture, release 2. In *Journal of the American Medical Informtics Association*, volume 13, pages 30–39, Kaiser Permanente, Pasadena, CA, USA. robert.h.dolin@kp.org, 2006.

[DC99] Steven DeRose and James Clark. XML Path Language (XPath) Version 1.0. W3C, http://www.w3.org/TR/1999/REC-xpath-19991116, 1999.

[Dep06] Department of Inpatient Nursing – The Ohio State University Medical Center. Intravenous (IV) Therapy. http://medicalcenter.osu.edu/pdfs/PatientEd/Materials/PDFDocs/procedure/invas-pr/intra-ve.pdf, 2006.

[FW] David C. Fallside and Priscilla Walmsley. XML Schema Part 0: Primer Second Edition. World Wide Web Consortium Recommendation. http:www.w3.org/TR/xmlschema-0/, 2004.

[Gar] Sebastian Garde. An Introduction to Developing and Managing Archetypes. http://healthinformatics.cqu.edu.au/projects/openEHR/openEHRArchetypesTutorial.pdf.

[GHBK07] Sebastian Garde, Evelyn Hovenga, Jasmin Buck, and Petra Knaup. Expressing clinical data sets with openEHR archetypes: A solid basis for ubiquitous computing. In *International Journal of Medical Informatics*, volume 76, pages 334–341. Elsevier, 2007.

[GKHH07] Sebastian Garde, Petra Knaup, Evelyn J.S. Hovenga, and Sam Heard. Towards semantic interoperability for electronic health records: domain knowledge governance for openehr archetypes. In *Methods of Information in Medicine*, volume 46, pages 332–343. Schattauer, 2007.

[GL96] François Grémy and Jean Lelaidier. Is there anything new about the so-called ”medical” record? In *Methods of Information in Medicine*, volume 35, pages 93–107. Schattauer, 1996.

[GoHSA08] Victorian State Government and Department of Human Services Australia. An Overview of Public Hospital Activity. http://www.health.vic.gov.au/yourhospitals/emergency/index.htm, 2008.

[HM04] Elliotte Rusty Harold and Scott Means. *XML in a Nutshell*. O’Reilly Media, Sebastopol, CA, USA, 3rd edition, 2004.

[Inf06] Ocean Informatics. CEN, CDA and openEHR. http://oceaninformatics.biz/CMS/index.php?option=com_content&task=view&id=31&Itemid=34, 2006.

[ISO05] ISO. Health informatics – Electronic health record – Definition, scope, and context. International Organization for Standardization, Geneva, 2005.

[JRH] Ian Jacobs, David Raggett, and Arnaud Le Hors. HTML 4.01 specification. World Wide Web Consortium Recommendation, December. http://www.w3.org/TR/1999/REC-html401-19991224, 1999.

[KGa05] Dräger Medical AG & Co. KGaA. *MEDIBUS for Dräger Intensive Care Devices*. Lübeck, 11th edition, 2005.

[Les] Heather Leslie. OpenEHR: The World’s Record. *Pulse+IT Magazine*. http://www.pulsemagazine.com.au/index.php?option=com_content&task=view&id=239&Itemid=1, 2007.

[MFBH07] Chunlan Ma, Heath Frankel, Thomas Beale, and Sam Heard. EHR Query Language (EQL) – A Query Language for Archetype-Based Health Records. In *MEDINFO 2007*, volume 12, pages 397–401, Amsterdam, 2007. IOS.

[N0̈7] Jens Nödler. An XML-based Approach for Software Analysis Applied to Detect Bad Smells in TTCN-3 Test Suites. Master’s thesis, Institute for Informatics, ZFI-BM-2007-36, 1612-6793, Center for Informatics, University of Göttingen, 2007.

[neh06] nehta. Review of Shared Electronic Health Record Standards. http://www.nehta.gov.au, 2006.

[oF] The openEHR Foundation. Introducing openEHR. http://www.openEHR.org.

[oF06] openEHR Foundation. ISO Standards, CEN Standards, HL7 Standards. http://www.openehr.org/standards, 2006.

[oF07] openEHR Foundation. openEHR Primer. http://www.openehr.org/shared-resources/openehr_primer.html, 2007.

[PPTK05] Lise Poissant, Jennifer Pereira, Robyn Tamblyn, and Yuko Kawasumi. The impact of electronic health records on time efficiency of physicians and nurses: A systematic review. In *Journal of the American Medical Informatics Association*, volume 12, pages 505–516. Elsevier, 2005.

[Sch04] Peter Schloeffel. Current EHR Developments: an Australian and International Perspective. Health Care and Informatics Review Online, http://hcro.enigma.co.nz/website/index.cfm?fuseaction=articledisplay&FeatureID=010904, 2004.

[SGHB06] Thilo Schuler, Sebastian Garde, Sam Heard, and Thomas Beale. Towards Automatically Generating Graphical User Interfaces from openEHR Archetypes. In *Studies in Health Technology and Informatics*, volume 124, pages 221–226. IOS, 2006.

[SHBR99] Peter Schoeffel, Sam Heard, Thomas Beale, and David Rowed. Overview of GEHR from an Australian Perspective. http://www.omg.org/docs/corbamed/00-01-17.pdf, 1999.

[vG95] Astrid M. van Ginneken. The structure of data in medical records. In *Yearbook of Medical Informatics 1995*, pages 61–70. Schattauer, 1995.

[WPJ+05] Jan Walker, Eric Pan, Douglas Johnston, Julia Adler-Milstein, David W. Bates, and Blackford Middleton. The Value Of Health Care Information Exchange And Interoperability. In *Health Tracking*, pages 10–18, 2005.

All URIs mentioned in this thesis have been verified on the 2008-05-12. Copies of the pages can be requested by sending an email to goek.murat@gmail.com.

List of Figures

List of Tables

Appendix A

Additional Information

A.1 Matching Tables

These tables represent some data items from the paper based forms of the emergency department and the corresponding openEHR archetypes.

Intravenous Therapy M 100	*Archetypes*
Line No Flask No Type of Fluid Vol. of Flask (ml) Additives including Dose Rate Date of Admin. Time started Time finished	openEHR-EHR-INSTRUCTION.medication.v1 with openEHR-EHR-ITEM_TREE.medication-formulation.v1

Table A.1: Data items - archetypes matching table.

Drug Therapy M 85.5	*Archetypes*
Date Route Medication Frequence Drug Level Time Level Taken Dose	openEHR-EHR-INSTRUCTION.medication.v1 with openEHR-EHR-ITEM_TREE.medication.v1

Table A.2: Data items - archetypes matching table.

A.2 Dräger MEDIBUS Commands

This table represent an extract of the commands for the Dräger ventilation system interface.

Code	*Command-Specification*
30H	Do nothing (NOP)
49H	Time changed
51H	Initialize Communication (ICC)
52H	Request Device Identification

Table A.3: Dräger Savina – transmitted commands.

Code	*Command-Specification*
24H	Request current DATA
25H	Request current LOW ALARM LIMITS
26H	Request current HIGH ALARM LIMITS
27H	Request current ALARMS (Codepage 1)
28H	Request Current Date And Time
29H	Request current DEVICE SETTINGS
2AH	Request current TEXT MESSAGES
2EH	Request current ALARMS (Codepage 2)
30H	Do nothing (NOP)
4AH	Configure Data Response
51H	Initialize Communication (ICC)
52H	Request Device Identification
53H	Request Realtime Configuration
54H	Configure Realtime Transmission
55H	Stop Communication
6AH	Device Specific

Table A.4: Dräger Savina – processed and responded commands.

A.3 RS232 Interface

This is a screenshot of the program for the RS232 interface for the Dräger Savina ventilation system.

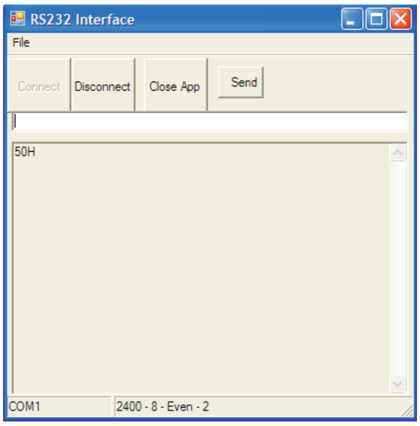

Figure A.1: Screeshot of the RS232 interface.

A.4 Template Data Schema

This listing represents the template data schema (tds) for the example used in this thesis. The tsd of the entire monitoring template can be found on the attached CD-ROM

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema xmlns:oe="http://schemas.openehr.org/v1" xmlns="http://schemas.
    oceanehr.com/templates/" elementFormDefault="qualified" targetNamespace="
    http://schemas.oceanehr.com/templates/" xmlns:xs="http://www.w3.org/2001/
    XMLSchema">
  <xs:import schemaLocation="Structure.xsd" namespace="http://schemas.openehr.org
      /v1" />
  <xs:element name="Monitoring">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name">
          <xs:complexType>
            <xs:sequence>
              <xs:element default="New Template" name="value" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element minOccurs="0" maxOccurs="1" name="uid" type="oe:UID_BASED_ID"
             />
        <xs:element name="language">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="terminology_id">
                <xs:complexType>
```

```
          <xs:sequence>
            <xs:element fixed="ISO_639-1" name="value" type="xs:token" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element fixed="en" name="code_string" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="category">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="value">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="event" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="defining_code">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="terminology_id">
              <xs:complexType>
                <xs:sequence>
                  <xs:element fixed="openehr" name="value" type="xs:token" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element fixed="433" name="code_string" type="xs:string" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="composer" type="oe:PARTY_PROXY" />
<xs:element minOccurs="0" maxOccurs="unbounded" name="Non-drug_therapy">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="name">
        <xs:complexType>
          <xs:sequence>
            <xs:element default="Non-drug therapy" name="value" type="xs:string" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="1" name="uid" type="oe:UID_BASED_ID" />
      <xs:element name="subject" type="oe:PARTY_SELF" />
      <xs:element minOccurs="0" maxOccurs="1" name="provider" type="oe:PARTY_IDENTIFIED" />
      <xs:element minOccurs="0" maxOccurs="unbounded" name="other_participations" type="oe:PARTICIPATION" />
      <xs:element name="narrative" type="oe:DV_TEXT" />
      <xs:element minOccurs="0" maxOccurs="1" name="new_activity">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="name">
              <xs:complexType>
```

```
    <xs:sequence>
      <xs:element default="new activity" name="value" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element minOccurs="0" maxOccurs="1" name="uid" type="oe:UID_BASED_ID" />
<xs:element name="timing" type="oe:DV_PARSABLE" />
<xs:element minOccurs="0" maxOccurs="unbounded" name="Gas_administration">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="name">
        <xs:complexType>
          <xs:sequence>
            <xs:element default="Gas administration" name="value" type="xs:string" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="1" name="Amount">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="name">
              <xs:complexType>
                <xs:sequence>
                  <xs:element default="Amount" name="value" type="xs:string" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element minOccurs="0" maxOccurs="1" name="Inspired_oxygen">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="name">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element default="Inspired oxygen" name="value" type="xs:string" />
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                  <xs:element name="value" type="oe:DV_PROPORTION" />
                </xs:sequence>
                <xs:attribute fixed="at0002" name="archetype_node_id" type="extendedArchetypeNodeId" use="required" />
                <xs:attribute fixed="oe:ELEMENT" name="type" use="required" />
              </xs:complexType>
            </xs:element>
            <xs:element minOccurs="0" maxOccurs="1" name="Flow_rate">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="name">
                    <xs:complexType>
                      <xs:sequence>
```

```
                  <xs:element default="Flow rate" name="value" type="xs:string" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="value" type="oe:DV_QUANTITY" />
          </xs:sequence>
          <xs:attribute fixed="at0005" name="archetype_node_id" type="extendedArchetypeNodeId" use="required" />
          <xs:attribute fixed="oe:ELEMENT" name="type" use="required" />
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="1" name="Partial_pressure_of_oxygen">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="name">
              <xs:complexType>
                <xs:sequence>
                  <xs:element default="Partial pressure of oxygen" name="value" type="xs:string" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="value" type="oe:DV_QUANTITY" />
          </xs:sequence>
          <xs:attribute fixed="at0012" name="archetype_node_id" type="extendedArchetypeNodeId" use="required" />
          <xs:attribute fixed="oe:ELEMENT" name="type" use="required" />
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute fixed="at0004" name="archetype_node_id" type="extendedArchetypeNodeId" use="required" />
    <xs:attribute fixed="oe:CLUSTER" name="type" use="required" />
  </xs:complexType>
</xs:element>
<xs:element minOccurs="0" maxOccurs="1" name="Mode">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="name">
        <xs:complexType>
          <xs:sequence>
            <xs:element default="Mode" name="value" type="xs:string" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="value">
        <xs:complexType>
          <xs:sequence>
```

```
                                      <xs:element name="value" type="xs:string" /
                                          >
                                      <xs:element minOccurs="0" maxOccurs="1"
                                          name="defining_code" type="
                                          oe:CODE_PHRASE" />
                                    </xs:sequence>
                                  </xs:complexType>
                                </xs:element>
                              </xs:sequence>
                              <xs:attribute fixed="at0016" name="
                                  archetype_node_id" type="
                                  extendedArchetypeNodeId" use="required" />
                              <xs:attribute fixed="oe:ELEMENT" name="type" use="
                                  required" />
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                        <xs:attribute fixed="openEHR-EHR-ITEM_TREE.
                            gas_administration.v1draft" name="archetype_node_id"
                            type="extendedArchetypeNodeId" use="required" />
                        <xs:attribute fixed="ITEM_TREE" name="type" use="required
                            " />
                      </xs:complexType>
                    </xs:element>
                    <xs:element fixed="" name="action_archetype_id" type="
                        xs:string" />
                  </xs:sequence>
                  <xs:attribute fixed="at0001" name="archetype_node_id" type="
                      extendedArchetypeNodeId" use="required" />
                  <xs:attribute fixed="oe:ACTIVITY" name="type" use="required" />
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute fixed="openEHR-EHR-INSTRUCTION.non_drug_therapy.v1draft
                " name="archetype_node_id" type="extendedArchetypeNodeId" use="
                required" />
            <xs:attribute fixed="oe:INSTRUCTION" name="type" use="required" />
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute fixed="openEHR-EHR-COMPOSITION.encounter.v1draft" name="
          archetype_node_id" type="extendedArchetypeNodeId" use="required" />
      <xs:attribute fixed="oe:COMPOSITION" name="type" use="required" />
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="extendedArchetypeNodeId">
    <xs:restriction base="xs:string">
      <xs:pattern value="([a-zA-Z][\w_]*-[\w][\w_]*-[\w][\w_]*\.[\w][\w_]*(-[\w
          ][\w_]*)*\.v\d+[a-zA-Z]*(\.\d)*)|(at(0\.[0-9]{1,4}|[0-9]{4})
          (\.[0-9]{1,3})*)" />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

Listing A.1: XML file example for showing the structure.

Appendix B

Contents of the CD-ROM

The CD-ROM which is attached to the print version of this thesis contains:

- A digital PDF version of this thesis.
- The templates as oet and the corresponding template data schemas (tds) as xsd:
  - For the monitoring of the vital signs.
  - For the intravenous therapy - order/ administration.
  - For the drug therapy.
- Source code and executeable file of the vb.net RS232 interface.
- A figure of the entire monitoring template data schema for the ventilation system as map.