USING THE DUAL-LEVEL MODELING APPROACH TO DEVELOP APPLICATIONS FOR PERVASIVE HEALTHCARE

JOÃO LUÍS CARDOSO DE MORAES[1], WANDERLEY LOPES DE SOUZA[1], LUÍS FERREIRA PIRES[2],
LUCIANA TRICAI CAVALINI[3] AND ANTÔNIO FRANCISCO DO PRADO[1]

[1] *Federal University of São Carlos, Computer Science, São Carlos, Brazil*
{joao_moraes, desouza, prado}@dc.ufscar.br
[2] *University of Twente, Software Engineering Group, Enschede, The Netherlands*
l.ferreirapires@utwente.nl
[3] *Rio de Janeiro State University, Department of Health Information Technology, Brazil*
lutricav@lampada.uerj.br

Health information technology is the area of IT involving the design, development, creation, use and maintenance of information systems for the healthcare industry. Automated and interoperable healthcare information systems are expected to lower costs, improve efficiency and reduce error, while also providing better consumer care and service. Pervasive Healthcare focuses on the use of new technologies, tools, and services, to help patients play a more active role in the treatment of their conditions. Pervasive Healthcare environments demand a huge amount of information exchange, and specific technologies have been proposed to provide interoperability between the systems that comprise such environments. However, the complexity of these technologies makes it difficult to fully adopt them and to migrate Centered Healthcare Environments to Pervasive Healthcare Environments. Therefore, this paper proposes an approach to develop applications in the Pervasive Healthcare environment, through the use of dual-level modeling based on Archetypes. This approach was demonstrated and evaluated in a controlled experiment that we conducted in the cardiology department of a hospital located in the city of Marília (São Paulo, Brazil). An application was developed to evaluate this approach, and the results showed that the approach is suitable for facilitating the development of healthcare systems by offering generic and powerful capabilities.

Key words: Pervasive Healthcare, Ubiquitous Computing, *open*EHR, Dual-level Modeling, Archetypes, Domain-Specific Language

1 Introduction

Most countries face many problems related to healthcare, such as expensive care and the low quality of health services. These problems began to appear as a result of population increase and the lack of healthcare professionals, which is worsening due to the increasing number of the elderly, more chronic disease, and the growing demand for new treatments and technologies. For example, in the United States the number of graduated doctors was 15,632 in 1981 and 15,712 in 2001 (increasing by only 0.5%), while in the same period its population rose from 226 million to 281 million (increasing by 24%) [1].

The current healthcare model is centered on employing highly specialized people, located in large hospitals and focusing on acute cases for treatment but needs to change into a distributed model, to produce faster responses and to allow patients to better manage their own health. The centralized healthcare model implies that patients and caregivers have to visit the same facility (a hospital or clinic) for the healthcare services to be delivered, which is often expensive and inefficient. A distributed healthcare model that pervades the daily lives of the citizens is more able to provide less

expensive and more effective and timely healthcare, and characterizes Pervasive Healthcare. According to [2], the goal of Pervasive Healthcare is to enable the management of health and well-being by using information and communication technologies to make healthcare available anywhere, at anytime, and to anyone.

Ubiquitous Computing [3] is a computing paradigm that aims at enabling the user to easily access and process information from anywhere, at anytime, and using any kind of device. Ubiquitous Computing environments, whether in communities, homes, or hospitals, can be extremely useful to build a Pervasive Healthcare model. Particularly, for the core of this model to become directed towards the patient, the efficient care of that patient is fundamental. In this manner, it is necessary that the information exchange among the various professionals, doctors, nurses and health experts responsible for patients is fast, efficient and safe.

The exchange of health information among heterogeneous Electronic Healthcare Record (EHR) systems in pervasive healthcare environments requires communication standards that enable interoperability between such systems. Although Health Level Seven (HL7)[a] is a widely-used international standard for message exchange between heterogeneous Healthcare Information Systems (HIS), it has some well-known limitations for representing clinical knowledge, such as its combined use of structured components and coded terms, which can result in inconsistent interpretations of clinical information [4].

HL7v3 is not a restriction-based standard, but it is XML-based, which could bring some advantages over a Domain-Specific Language (DSL), because of the ubiquity of XML technologies for data exchange and messaging [5]. However, achieving semantic interoperability with HL7v3 is challenging, since it is not fully restriction-based; therefore, there is no validity chain to insure conformance back to a known valid model. HL7v3 Reference Information Model (RIM)-based data models are all independently designed, as it can be seen by the update, simplification and extension that have taken place through its history, which poses maintenance issues for HL7v3-based applications [6].

Although the HL7v3 Clinical Document Architecture (CDA) [7] has been partially adopted as a reference document and there is now a tendency to use it as a base reference model, it is too large and has unnecessary requirements for many applications, such as, e.g., that mobile applications or devices can only use a push data approach [8]. Another recent initiative in the HL7 ecosystem is the Fast Healthcare Interoperability Resources (FHIR)[b], an attempt to open source some of the aspects of HL7 [9]. Once again, the FHIR specification lacks a common information model to validate data instances created according to their XML Schemas, which are centrally defined and imposed to the community of users, which ends up extending the schemas to accommodate their specific needs, thus breaking the semantic interoperability chain.

Other standards revolving around the HL7 system, such as the Integrating the Healthcare Enterprise (IHE)[c] profiles and the Standards and Interoperability (S&I) Framework[d] also promote

---

[a] http://www.hl7.org
[b] http://www.hl7.org/implement/standards/fhir/
[c] http://www.ihe.net/
[d] http://www.siframework.org/

semantic interoperability capabilities. The evaluation of those initiatives reveal that the IHE profiles actually define data structures that are implementable in *open*EHR archetypes, although the majority of IHE work is based around standardized workflow, which is not a semantic interoperability issue, being solved at the application implementation level [10]. The Standards and Interoperability (S&I) Framework is analogous to the HL7v3 CDA and the NCI CDE[e] initiatives, in that it can be defined as a top-down, document-centric approach that aims at gaining consensus on modeling concepts. The documents available from the S&I Content Browser can also be modeled as *open*EHR archetypes [11]. *open*EHR[f] is a foundation dedicated to research into interoperable EHR, which defined an open architecture based on two-level modeling that separates information from knowledge, therefore addressing some of the limitations of HL7 and its related standards [12].

This paper proposes an approach to developing applications in the Pervasive Healthcare environment, by using archetype and DSLs [13]. DSLs allow the creation of an infrastructure to reuse and generate most of the applications code of the target domain. The remainder of this paper is structured as follows. Section 2 introduces the main concepts of the *open*EHR standard. Section 3 describes the DSL concept. Section 4 presents the proposed approach illustrated with a case study in the healthcare domain. Section 5 shows the evaluation of the application developed using the proposed approach. Section 6 discusses some related work. Finally, section 7 presents concluding remarks and proposes further work.

2   *open*EHR Dual-Model

The *open*EHR architecture was developed based on the two-level modeling paradigm, as shown in Figure 1. At the first level, a common *Reference Model* (RM) was defined in terms of a predefined set of classes that model the structure of an electronic record; and on the second level, specific concepts were defined by restricting the RM classes in terms of so-called *archetypes*, expressed in the Archetype Definition Language (ADL) [14].
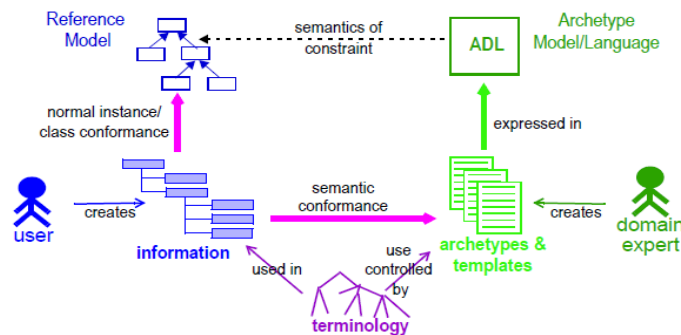

Figure 1 *open*EHR two-level modeling paradigm.

An archetype constitutes a formal model of a domain concept and is expected to be easily understandable by a domain expert. At the implementation level, archetypes can be translated into any

---

[e] https://cdebrowser.nci.nih.gov/
[f] http://www.openehr.org

language. In accordance with the two-level modeling, data input from users are stored according to the RM, but should also comply with the concepts expressed by the archetypes. The archetypes are designed by the domain experts, and not by information technologies professionals. This split should facilitate the interpretation of the knowledge extracted from the messages exchanged by health systems in various applications.

In the *open*EHR RM, the COMPOSITION class refers to one or more instances of the SECTION class, each containing ENTRY objects. The ENTRY class represents the actual recording of clinical content during a patient Observation, Examination, Assessment, or Intervention. ENTRY is defined as an abstract type with four concrete subtypes: (1) OBSERVATION, which can be used to represent clinical observations, such as blood pressure; (2) EVALUATION, which can be used to represent assessments made after a clinical observation is completed, such as risk assessment; and (3) INSTRUCTION and ACTION, which can be typically used to represent surgical procedures, medication, and other clinical interventions and actions taken. The ACTION subclass describes what was done and committed to the EHR as the result of an INSTRUCTION.

The Clinical Knowledge Manager (CKM)[g] is the archetype repository proposed by the *open*EHR Foundation. This repository contains a set of archetypes that represent clinical concepts and can be reused in various health applications. We have reused some available archetypes provided by CKM, such as *Device, Device Details* and *Clinical Synopsis*, but we have also developed new archetypes to represent clinical concepts of the cardiology domain, such as *Pacemaker Implantation, Vascular Cardiac Surgery, Coronary Cardiac Surgery, Angioplasty Cardiac* and *Pacemaker Evaluation*.

```
1   archetype (adl_version=1.4) openEHR-EHR-ACTION.pacemaker_implantation.v1
2   [...]
3   definition
4     ACTION[at0000] matches {--Pacemaker Implantation
5       description matches {
6         ITEM_TREE[at0001] matches {   -- Tree
7           items cardinality matches {2..*; unordered} matches {
8             CLUSTER[at0008] occurrences matches {0..1} matches {--Stimulations Mode
9               items cardinality matches {1..2; unordered} matches {
10                ELEMENT[at0003] occurrences {0..1} matches {--Stimulation Mode
11                  value matches {
12                    DV_TEXT matches {*}}}
13  [...]
14  ontology
15    term_definitions = <
16      ["en"] = <
17        ["at0000"] = <
18          text = <"Pacemaker Implantation">
19          description = <"The Pacemaker implant is indicated for patients[...]">
20  [...]
```

Figure 2 Archetype for the Pacemaker Implantation concept.

The *open*EHR archetypes were defined for general reuse, and they can accommodate any number of natural languages and terminologies. An archetype consists of three sections: header, definition and ontology. Figure 2 shows an extract of an ADL archetype for *Pacemaker Implantation* defined according to the *open*EHR standard. The *header* includes the name of the archetype (line 1). The

---

[g] http://www.openehr.org/ckm/

definition section contains the structure and restrictions associated with the clinical concept defined by the archetype. *Pacemaker Implantation* specializes the ACTION class (line 4) of the RM. The CLUSTER part (line 8) refers to the stimulations mode for pacemaker implantation, and consists of an ELEMENT with a value of type DV_TEXT. The *ontology* section (line 14-20) includes the terminological definitions. In this example, the linguistic expression 'Pacemaker Implantation' is associated with the code 'at0000'.

The two-level modeling approach applied in *open*EHR allows designers to develop software systems separately from the domain modeling, by specializing and instantiating RM classes. Domain experts can introduce new concepts by defining new archetypes, so that the software systems do not have to be redesigned and redeployed whenever a new concept is defined. Therefore, the two-level modeling approach tends to make the systems easier to maintain and extend with new applications. Archetypes play the role of semantic gateways to terminologies, classifications and computerized clinical guidelines. Archetypes were introduced into *open*EHR to improve the level of semantic interoperability between various healthcare information systems.

3   Domain-Specific Languages

A DSL is a language designed to be useful to a specific set of tasks within a given domain [15]. It can be defined by a metamodel, which represents the domain knowledge of the target problem. Restricted to a specific domain, DSLs are usually small, consisting of just a set of abstractions and notations, which are closed to the terms known by the experts of this domain [16]. Thus, DSLs express solutions in the language and abstraction level of the problem domain, reducing the translation efforts of the concepts of this domain into the solution domain.

The use of DSLs in applications modeling, rather than general-purpose modeling languages (e.g., UML), allows the creation of more specific and complete models. Resources, such as frameworks, design patterns and components, may be included in models, creating an infrastructure that allows the execution of Model-to-Code (M2C) transformations to generate a greater amount of code from models.

There are some conceptual similarities between dual-level modeling and metamodeling, but there are also some important differences. The metamodeling approach is concerned with the overall architecture and development of a specific software application or specific system of applications developed around a set of requirements. This approach improves software quality and ease of maintenance. These are generally implemented using a DSL and a specific technology platform [17]. Although dual-modeling incorporates those same advantages, it extends the metamodeling approach to achieve semantic interoperability at the conceptual level, across every development platform. DSLs provide a significant level of power and control when used in a closed environment, and the *open*EHR specifications attempt to extend this to the healthcare ecosystem, where a multitude of software and hardware platforms must be accommodated [18].

4   Proposed Approach

In our approach, we applied *open*EHR dual-level modeling, as a model-driven methodology that is known to be suitable to achieve semantic interoperability. The approach combines archetypes and

templates in order to create communication interfaces, using ADL as the DSL to express the healthcare domain knowledge.

4.1. Overview

Using the Structured Analysis and Design Technique (SADT) [19] diagram notation, Figure 3 shows an overview of the approach, which consists of two stages: Domain Engineering (DE) and Application Engineering (AE). In DE, a DSL represented by a healthcare domain metamodel and Model-to-Code (M2C) transformations are defined. In AE, ubiquitous applications are developed through the reuse of the artifacts developed in the DE stage. The metamodel is used to support application modeling, and the M2C transformations are employed to generate the code that handles *open*EHR messages using archetypes.
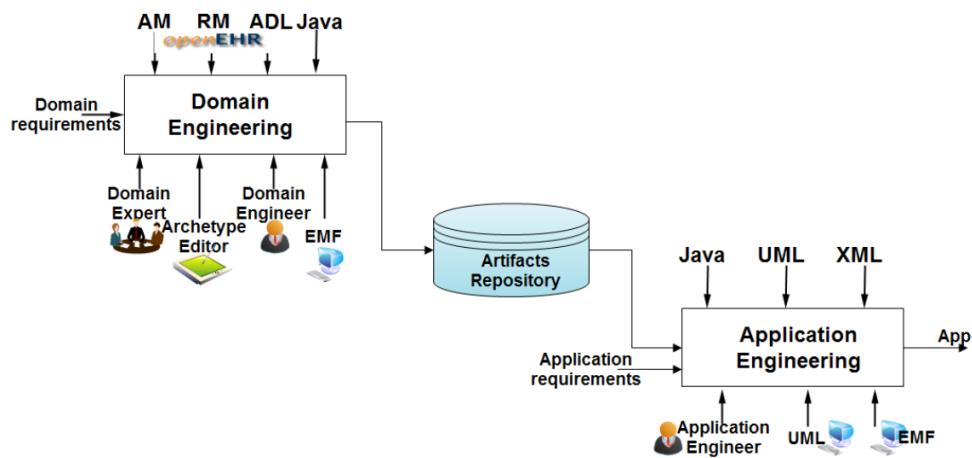

Figure 3 Overview of the proposed approach.

4.2. Domain Engineering

Figure 4 illustrates the three activities of the Domain Engineering stage: *Specify Domain Metamodel, Design Domain Metamodel,* and *Implement Domain Metamodel*.

In the **Specify Domain Metamodel** activity, the healthcare domain requirements are elicited, specified, analyzed and represented in a metamodel that expresses the knowledge about this domain. In the Eclipse Integrated Development Environment (IDE), the Domain Engineer, guided by the *open*EHR RM specifications, analyses the metamodel. The *open*EHR RM represents the healthcare domain information at a high abstraction level, which allows the use of a range of terminology in the concept description of this domain. The Domain Engineer specifies the archetypes, guided by *open*EHR AM specifications, ADL and the CKM repository. The Domain Experts and the Archetype Editor tool[h] are the main mechanisms used to support archetype specification. The outputs of domain analysis are domain metamodels, representing the concepts used in domain modeling, and the archetypes specified.
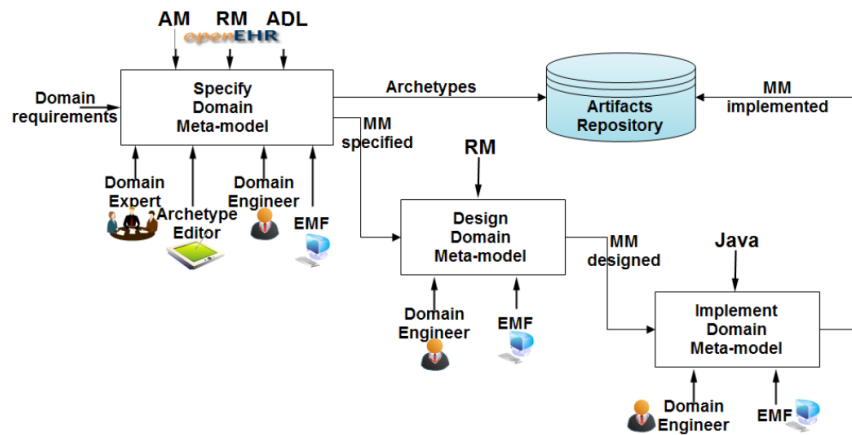
---

[h] http://www.oceaninformatics.com/

Figure 4 Domain Engineering overview.

In the ***Design Domain Metamodel*** activity, the metamodel analysis is refined according to standards, technologies and platforms that enable metamodel construction. The Domain Engineer uses the Eclipse EMF (Eclipse Modeling Framework) plugins [20] to develop the architecture for the domain metamodel, showing the functional decomposition of the domain. The process is guided by medical terminology coding, such as the Systematized Nomenclature of Medicine – Clinical Terms[i] (SNOMED-CT). SNOMED-CT is a scheme for identifying clinical terms and concepts, and is used by computer systems to support semantic interoperability. Figure 5 shows part of the designed metamodel with a flexible structure, which satisfies all relevant requirements and still leaves a large degree of freedom for its implementation.
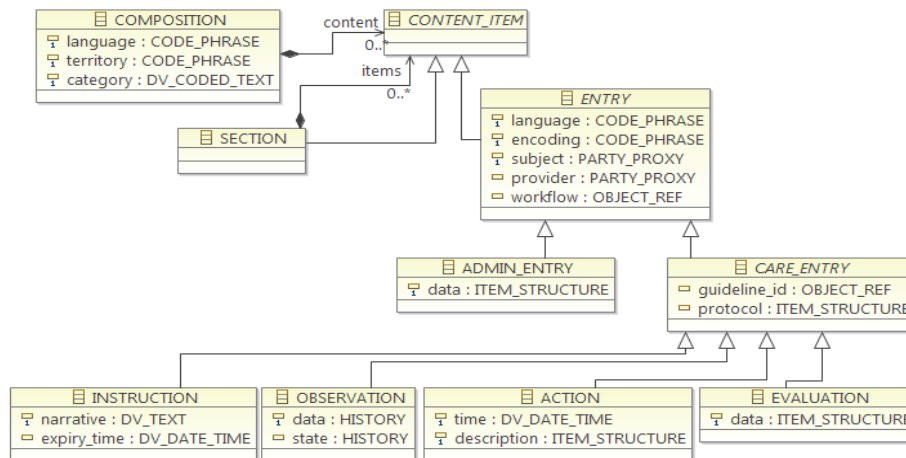

Figure 5 Portion of *open*EHR RM meta-model.

In the ***Implement Domain Metamodel*** activity, code is generated based on the designed metamodel. Supported by the EMF framework, the Domain Engineer automatically generates Java code for the

---

[i] http://www.ihtsdo.org/snomed-ct/

metamodel elements and for a model editor. Figure 6 shows the Java code generated (2) for the ACTION class, based on the designed metamodel (1). This model editor assists the Domain Engineer during the application model specification, in the AE stage. Both the model editor and the metamodel are available as Eclipse plugins, ensuring that during AE the metamodel is instantiated when the application is modeled. Since the metamodel is based on *open*EHR RM, the models created using a DSL expressed by this metamodel, can be considered as a refinement performed to create the *open*EHR messages.
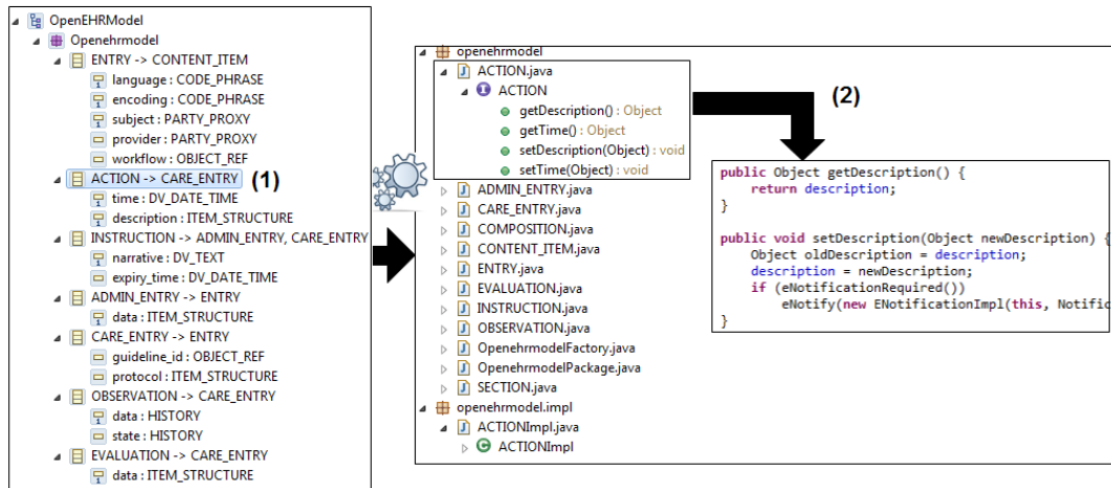

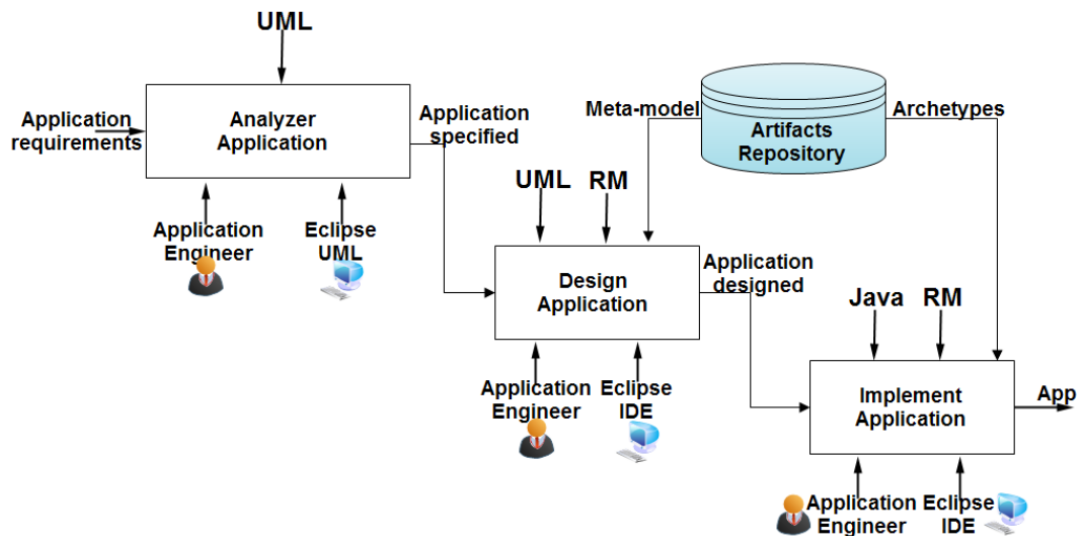Figure 6 Sketch of metamodel with Java code generation.

Figure 7 Application Engineering overview.

4.3. Application Engineering

The AE stage involves the disciplines of Analysis, Design, and Implementation, which are part of the software development process. Figure 7 shows the three activities of the Application Engineering stage: *Analyzer Application, Design Application,* and *Implement Application*.

Besides facilitating application development, the use of a DSL expressed by a healthcare domain metamodel to application modeling in AE, allows the mapping of the development process for *open*EHR messages onto Software Engineering activities. Thus, the *open*EHR RM can be transformed directly to models that comply with the application requirements. Moreover, the code generation automates many of the Application Engineer tasks associated with *open*EHR messages, complying with the concepts expressed by the archetypes. The healthcare metamodel also allows the reuse of domain knowledge in various projects during the AE.

To investigate the feasibility of the proposed approach, a case study was developed in which a ubiquitous application was built that allows the evaluation of an implanted pacemaker in patients. The patient receives a notification on his mobile device to schedule an appointment. When the patient arrives at the clinic, the physician receives a message on his device containing information about the patient's pacemaker implantation. The messages received by the physician and the patient are based on *open*EHR archetypes related to the pacemaker implantation concept. The AE activities are detailed below, illustrating the application developed in the case study.

In the ***Analyzer Application*** activity, the application is specified from its requirements, which have been elicited, specified, analyzed and verified. For the specification, the Application Engineer uses UML techniques such as class, sequence and use cases diagrams. An example is the use case diagram shown in Figure 8, which specifies the requirements to evaluate pacemaker implantation during an appointment. The clinical information was sent to the physician using the *open*EHR standard with archetypes.
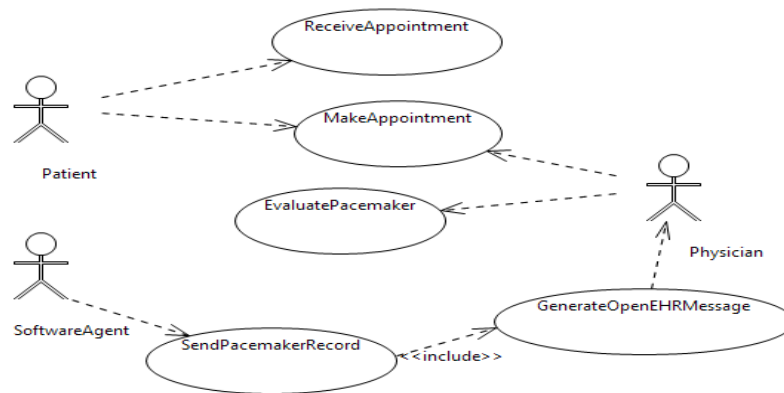

Figure 8 Application Use Cases.

In the ***Design Application*** activity, the application specification is refined using the technologies of hardware and software platforms that enable the implementation of the application, such as Java Micro Edition (Java ME). Based on the use case diagrams, in this activity, the Application Engineer also

models the terms and concepts of the target problem domain as instances of the metamodel built in the DE stage, selecting the classes and attributes that are relevant to the application domain. Figure 9 shows a model built using the model editor plugin available in Eclipse.

This model describes the information associated with the domain applications in accordance with the pacemaker implantation archetype. The *PacemakerImplantation* class is a subtype of the ACTION class from *open*EHR RM, as shown in Figure 9 (1). The attribute *stimulationMode* represents the mode of stimulation used during the pacemaker implantation, according to the at0003 node in the *pacemaker_implantation* archetype, as shown in Figure 9 (2).

Adopting the coding terminology used in the metamodel (SNOMED-CT), the Application Engineer combines the archetypes that define the clinical concepts represented in the models using the corresponding SNOMED-CT code. For example, Figure 9 shows the *stimulationMode* attribute of the *PacemakerImplantation* class, which was assigned the SNOMED-CT code that identifies the pacemaker implantation concept on the archetype ontology section. Thus, the application becomes able to transmit understandable clinical data in an interoperable way.
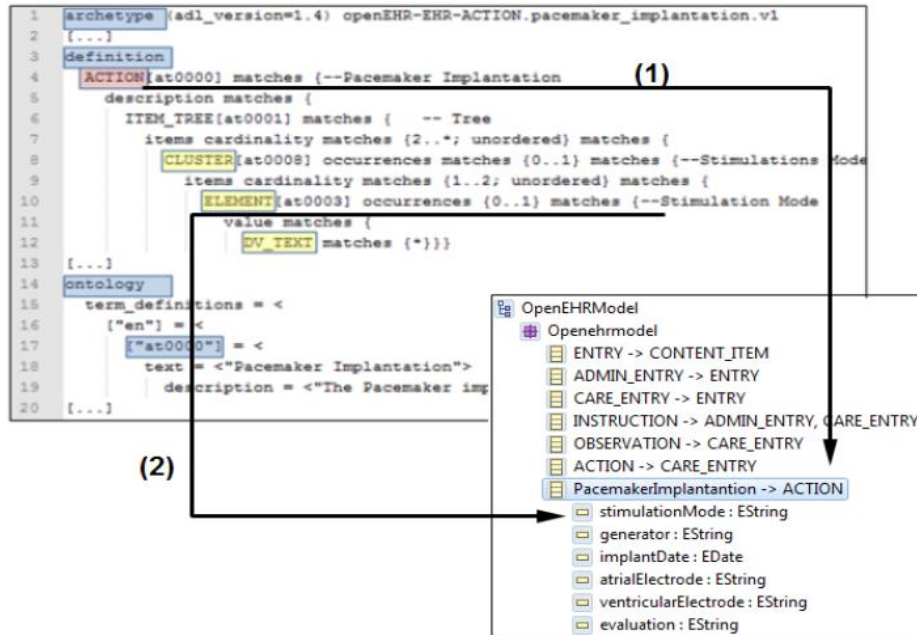

Figure 9 Meta-model of the Pacemaker Implantation concepts.

In the **Implementation Application** activity, the application is implemented together with its communication interface by integrating archetypes with *open*EHR messages. In Eclipse, the Application Engineer is assisted by the JET framework to perform the M2C transformations to generate partial code. This code handles the *open*EHR messages using archetypes, which carries the data related to pacemaker implantation, according to specifications contained in the model. After the partial code generation, the Application Engineer complements the code to make the application executable. Moreover, the other components that comprise the application are developed, such as the

user integration interfaces and data persistence. At the end of implementation, tests are performed to provide feedback that indicates whether it is necessary to return to the previous activities of the AE stage. Figure 10 shows the *pacemaker_implantation* archetype and the user interface that contain clinical information related to the case study.
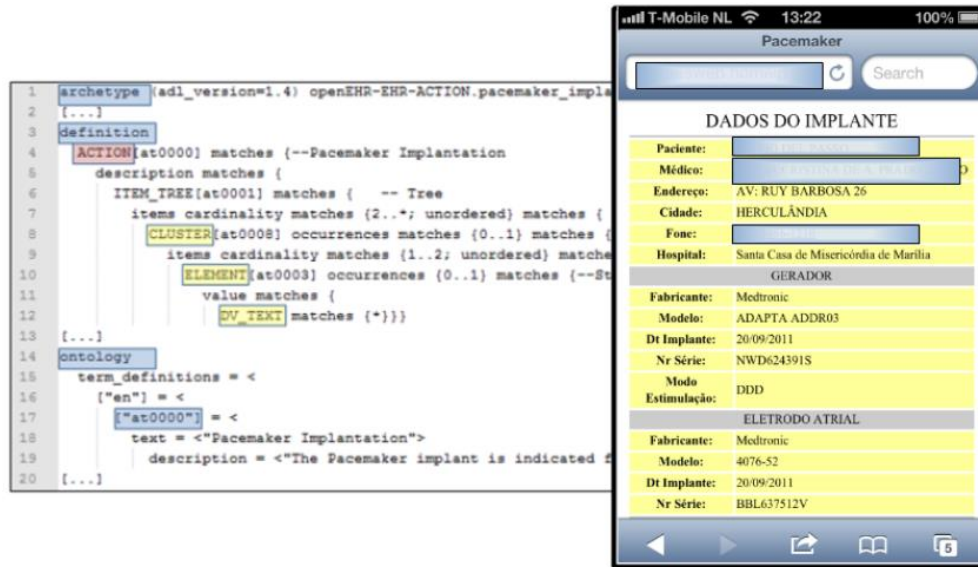

Figure 10 Pacemaker implantation – use case.

5   Results and Discussion

We conducted a controlled experiment [21] in the cardiology department of the Santa Casa Hospital of Marília (São Paulo, Brazil), using the scenario discussed in Section 4. Table 1 shows the number of participants. The evaluation period was from 1 January 2012 to 30 July 2012.

Table 1. Participants (respondents).

Participants	Physician	Medical Student	Nurse	Patient	Total
Scenario: Pacemaker Evaluation	3	2	2	15	22

In our study we used the Technology Acceptance Model (TAM) [22], which assumes that *perceived usefulness* (PU) and *perceived ease of use* (PEU) can predict attitudes towards usage and *intentions to use* (IU) technology. TAM is known to be a suitable model for explaining the technology acceptance process in the healthcare sector [23]. To test the effect on technology acceptance, two groups were selected for analysis, involving a limited but still relevant set of people: caregivers (physicians, medical students and nurses) and patients. We decided to group the caregivers together because they work as a team in the scenario.

After the implementation of the scenario, a structured questionnaire based on TAM was sent to the caregivers involved in the scenario, and a questionnaire was also distributed to patients after the

medical procedure. Respondents were asked to indicate their agreement or disagreement with several statements on a five-point Likert scale [24], ranging from 1=strongly disagree to 5=strongly agree. Many empirical studies have demonstrated that the psychometric properties of measurement scales can be affected by the ordering of items within the questionnaire. To avoid this bias, the questionnaire randomly intermixed items across constructs (PU, PEU, IU), and we conducted a group pretest to ensure that the scales were suitable. About 80% and 73% of the questionnaires distributed to caregivers and patients respectively were completed.

Our analysis consisted of two parts: (1) we tested the validity and reliability of the measurement model using Cronbach's Alpha [25]; (2) the data were analyzed using Structural Equation Modeling (SEM) [26] to test both the research model and the hypotheses. SEM is a statistical method that combines factor analysis and path analysis, enables theory construction and analyses the strengths of relationships among variables.

Table 2 shows the results of the validity tests for all scenarios, in which the internal consistency of the constructs were further evaluated for their reliability.

Table 2. Statistics and reliability of constructs.

Participants	Constructs	Number of Items	Mean	Cronbach's Alpha
Caregivers	Perceived Ease of Use	4	4.05	0.813
	Perceived Usefulness	5	4.50	0.851
	Intention to Use	3	4.39	0.799
Patients	Perceived Ease of Use	4	4.13	0.790
	Perceived Usefulness	5	4.06	0.862
	Intention to Use	3	3.94	0.764

The constructs had Cronbach's Alpha values at least close to the limit of 0.700, which is considered very good. Therefore, we concluded that these constructs are reliable for use in our data analysis. Based on this analysis, we concluded that the mean values showed good validity.

Next, the validated data were analyzed using SEM. The users' intentions to use the communication systems in each scenario can be explained or predicted based on their perception of ease of use and usefulness. Figure 11 summarizes the research model used in this study. The labels on the arrows show the hypotheses and the path coefficients that show the relative strength and together indicate the causal relationships among variables, whereas R^2 is the percentage of total variance of the independent variables and indicates the predictability of the research model.
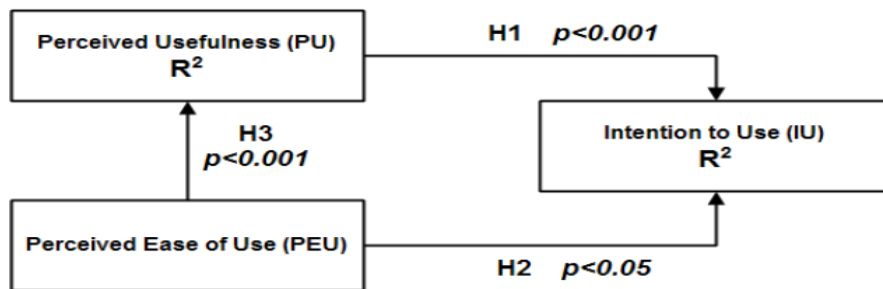

Figure 11 Research model and hypotheses.

The following three hypotheses were proposed and analyzed in this scenario:

H1. Perceived Usefulness positively affects Intention to Use the system.
- *Null hypothesis: H_{1n}: $\mu_{PerceivedUsefulness} = \mu_{IntentionUse}$.*
- *Alternative hypothesis: H_{1a}: $\mu_{PerceivedUsefulness} \neq \mu_{IntentionUse}$.*

H2. Perceived Ease of Use positively affects Intention to Use the system.
- *Null hypothesis: H_{2n}: $\mu_{PerceivedEaseUse} = \mu_{IntentionUse}$.*
- *Alternative hypothesis: H_{2a}: $\mu_{PerceivedEaseUse} \neq \mu_{IntentionUse}$.*

H3. Perceived Ease of Use positively affects Perceived Usefulness.
- *Null hypothesis: H_{3n}: $\mu_{PerceivedEaseUse} = \mu_{PerceivedUsefulness}$.*
- *Alternative hypothesis: H_{3a}: $\mu_{PerceivedEaseUse} \neq \mu_{PerceivedUsefulness}$.*

To verify these three hypotheses, we chose to apply *statistical regression analysis* [27] to the data collected from caregivers and patients in this scenario. The results are summarized in Table 3. For example, in the scenario (participants=caregivers), we test H_{3a} to verify if Perceived Usefulness is determined by Perceived Ease of Use. The details of the regression analysis are: R^2=0.61, p=0.0005 (very high significance with p<0.001 and α=0.05). From the result of the *regression* we could reject the null hypothesis (H_{3n}), meaning that we empirically corroborate that Perceived Usefulness is determined by Perceived Ease of Use, and that H_{3a} was strongly confirmed. R^2 indicates that Perceived Ease of Use explains 61% (R^2=0.61) of the variance in Perceived Usefulness. Hypotheses H_{1a} and H_{2a} were confirmed, and Perceived Usefulness strongly determined Intention to Use, and Perceived Ease of Use was a significant secondary determinant.

Table 3. Statistic regression analysis.

	Hypotheses	H1 (PU→IU)		H2 (PEU→IU)		H3 (PEU→PU)	
Scenario	Participants	R^2	p<0.001	R^2	p<0.05	R^2	p<0.001
Pacemaker Evaluation	Caregivers	0.72	0.0001	0.72	0.009	0.61	0.0005
	Patients	0.35	0.0000	0.35	0.007	0.55	0.0002

Therefore, we could conclude that all three hypotheses were confirmed at all points of measurement. Based on this evaluation, we can conclude that the caregivers think that the system would be very useful for daily tasks and was very easy to use. Most patients identified some usability benefits, such as the efficient message exchange.

However, our data analysis approach has some limitations. First, the questionnaire model is not completely free of subjectivity for each respondent. Each respondent reacts in a particular way to a questionnaire. Second, we grouped all caregivers together and generalized the results, while we could have split them in several groups. Third, other factors may affect the decision of people to use a given technology, such as their prior experience and job relevance [28], which we did not take into consideration in our work. However, in this study we considered perceived usefulness and perceived ease of use as the most important factors to explain future intention to use the system.

6   Related Work

Various proposals to facilitate the use of the electronic healthcare standards can be found in the literature, which have been motivated mainly by the complexity of the message development process and the lack of comprehensive tool support

In [29], the implementation of a modeling tool for HL7 messages developed in Eclipse is discussed. To handle the artifacts, the developed tool allows the serialization of the messages in the XML Metadata Interchange (XMI) standard for later use. However, for dual-level model, the metamodeling approach using EMF has showed that the EMF locks the developer into that technology, since even the XML Schema export process includes EMF dependencies. In addition to this, the Eclipse plugins did not fully support XML Schema 1.1, which is necessary for dual-level modeling, since multiple substitution groups are required to allow multiple constraints of the same Reference Model class in an archetype. Another XML Schema 1.1 required feature, yet not supported by EMF, is the inclusion of asserts in the schema, for internal validation of the constraints defined to the class attributes converted into *ComplexType* elements. To date, the lack of support for multiple substitution groups and assertions makes it impossible to export archetypes into a format that was usable outside of the EMF.

In [30], an approach is presented for developing laboratory information system (LIS) software by using archetypes to improve the software development process. This approach together with associated implementation of the clinical LIS software factory shows that archetypes and archetype patterns can be used as a basis to model the business domain.

In [31], the implementation of a software factory for the healthcare domain is discussed. The goal of developing the software factory was to automate the creation of communication interfaces, termed *collaboration ports*, in order to facilitate message exchange. In this approach to interoperability, the authors share the experience gathered in designing and implementing a software factory for healthcare systems based on the Health Level Seven (HL7) standard. The authors discuss the long-term vision and the scoped-down proof of concept developed so far, and also outline the challenges encountered in their project and the opportunities to widen the scope of the approach to different industries, and, in general, the opportunities to support business-to-business collaboration.

In [32], a solution is presented to assist developers in handling messages. This solution is based on the use of a programmatic DSL to support the messages creation, transmission, interpretation and validation, among other features.

In [33], the *open*EHR specifications are adopted to model administrative data related to the claim submissions issued by healthcare insurance companies. The software implementation showed that semantic interoperability in the selected domain could be achieved, although the archetypes created are not yet acknowledged by the inclusion and publication in the *open*EHR Foundation Clinical Knowledge Manager (CKM) archetype repository. This is required according to the current version of the *open*EHR specifications, in order to avoid semantic conflicts related to ADL file identification using archetype IDs.

In [34], the most complete migration process from a traditional healthcare information system to an *open*EHR based application is presented, including a detailed description of the stages adopted by the authors during the archetype modeling phase. The authors concluded that by developing the

knowledge layer and persistence layer and due to the current style of modeling in traditional healthcare information system, this migration is straightforward. Nevertheless, more effort is required for the other two layers, especially for the service layer for which implementations are not available.

The approach proposed in this paper is based on several characteristics of the works described above. It offers its own contributions through the evolution and adaptation of concepts described in related works. The use of archetypes to represent clinical concepts that are integrated with *open*EHR messages is one of the main contributions of this work. We also use a DSL based on a healthcare domain metamodel, which enables modeling and automatic code generation of the communication interfaces of the applications in that domain.

7   Conclusions

This paper has presented an approach to develop applications in the Pervasive Healthcare environment that integrates archetypes to *open*EHR messages via a DSL. The integration of archetypes to *open*EHR messages achieves agility and efficiency in communication between heterogeneous HIS, making these systems interoperable under the assumption that clinical concepts can be identified and formally represented. The use of a DSL allows the creation of an infrastructure for code generation of these messages, and their reuse in different application of target domain, which reduces the necessary application development effort.

We performed a comprehensive study to investigate the usefulness and ease of use of pervasive healthcare technologies within healthcare environments. We presented the scenario that shows the acceptance of our pervasive healthcare approach by both caregivers and patients, who reacted positively with respect to the usefulness of the architecture, and also to task delegation to intelligent agents. In our approach, the partial code for handling *open*EHR using archetypes messages was generated with the application of suitable M2C transformations.

In further work, we will further evaluate the performance of our approach, especially its scalability, which is a crucial non-functional requirement for realistic applications and for the simultaneous support of multiple scenarios. We will also measure any improvement in patient safety during the medical procedure. We plan future experiments to extend the TAM models to investigate the effect of other variables (e.g., user experience, job relevance and output quality) on perceived usefulness, perceived ease of use and intention to use. Modeling tools should also be improved so that domain experts can be better engaged in the development process. As EMF gains more maturity, we expect that it will become more adequate to support the metamodeling approach.

Acknowledgements

References

[1] U. Varshney, *Pervasive Healthcare Computing - EMR/EHR, Wireless and Health Monitoring*, pp. 282, New York: Springer, 2009.

[2] U. Hansmann *et al.*, *Pervasive Computing : The Mobile World*, 2nd Edition ed.: Springer, 2003.

[3] M. Weiser, "Some Computer Science Issues in Ubiquitous Computing," *Commun. ACM,* vol. 36, no. 7, pp. 75-84, 1993.

[4] E. Browne, *openEHR Archetypes for HL7 CDA Documents*, vol. 2012, Ocean Informatics, 2008.

[5] F. P. Coyle, "XML, web services and the changing face of distributed computing," *Ubiquity,* vol. 2002, no. April, pp. 2, 2002.

[6] M. Alam *et al.*, "Design and Implementation of HL7 V3 Standard-Based Service Aware System." pp. 420-425.

[7] R. H. Dolin *et al.*, "HL7 Clinical Document Architecture, Release 2," *Journal of the American Medical Informatics Association,* vol. 13, no. 1, pp. 30-39, Jan-Feb, 2006.

[8] A. Ryan, P. Eklund, and B. Esler, "Toward the Interoperability of HL7 v3 and SNOMED CT: A Case Study Modeling Mobile Clinical Treatment," *Medinfo 2007: Proceedings of the 12th World Congress on Health (Medical) Informatics, Pts 1 and 2,* vol. 129, pp. 626-630, 2007.

[9] D. Bender, and K. Sartipi, "HL7 FHIR: An Agile and RESTful approach to healthcare information exchange." pp. 326-331.

[10] D. S. Channin, "Integrating the healthcare enterprise: A primer - Part 6: The fellowship of IHE: Year 4 additions and extensions," *Radiographics,* vol. 22, no. 6, pp. 1555-1560, Nov-Dec, 2002.

[11] J. Kriseman *et al.*, "Harmonizing Content of Public Health Surveillance Systems: Lessons Learned from the ONC Standards and Interoperability (S&I) Public Health Reporting Initiative (PHRI)", 2013 CSTE Annual Conference, 2013.

[12] P. Schloeffel *et al.*, "The relationship between CEN 13606, HL7, and openEHR," in HIC 2006 Bridging the Digital Divide: Clinician, consumer and computer, 2006.

[13] M. Mernik, J. Heering, and A. M. Sloane, "When and how to develop domain-specific languages," *ACM Comput. Surv.,* vol. 37, no. 4, pp. 316-344, 2005.

[14] T. Beale, and S. Heard, *Archetype Definition Language*, openEHR Foundation, 2007.

[15] R. Gronback, *Eclipse Modeling Project: A Domain-Specific Language (DSL) Toolkit*, 1st ed., 2009.

[16] A. v. Deursen, P. Klint, and J. Visser, "Domain-specific languages: an annotated bibliography," *SIGPLAN Not.,* vol. 35, no. 6, pp. 26-36, 2000.

[17] A. Rutle *et al.*, "A metamodeling approach to behavioural modeling," in Proceedings of the Fourth Workshop on Behaviour Modeling - Foundations and Applications, Kgs. Lyngby, Denmark, 2012.

[18] C. Martinez-Costa *et al.*, "A model-driven approach for representing clinical archetypes for Semantic Web environments," *Journal of Biomedical Informatics,* vol. 42, no. 1, pp. 150-164, Feb, 2009.

[19] D. T. Ross, "Structured Analysis (SA): A Language for Communicating Ideas," *Software Engineering, IEEE Transactions on,* vol. SE-3, no. 1, pp. 16-34, 1977.

[20] D. Steinberg *et al.*, *EMF: Eclipse Modeling Framework 2.0*: Addison-Wesley Professional, 2009.

[21] A. R. Hevner *et al.*, "Design Science in Information Systems Research," *MIS Quarterly,* vol. 28, no. 1, pp. 75-105, Mar, 2004.

[22] F. D. Davis, "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology," *MIS Q.,* vol. 13, no. 3, pp. 319-340, 1989.

[23] P. Y. K. Chau, and P. J. H. Hu, "Investigating Healthcare Professionals' Decisions to Accept Telemedicine Technology: an Empirical Test of Competing Theories," *Information & Management,* vol. 39, no. 4, pp. 297-311, Jan, 2002.

[24] R. Likert, *A Technique for the Measurement of Attitudes*, New York: [s.n.], 1932.

[25] J. C. Nunnally, "Psychometric Theory," *Current Contents/Social & Behavioral Sciences*, no. 22, 1979.

[26] J. Henseler, and W. W. Chin, "A Comparison of Approaches for the Analysis of Interaction Effects Between Latent Variables Using Partial Least Squares Path Modeling," *Structural Equation Modeling-a Multidisciplinary Journal,* vol. 17, no. 1, pp. 82-109, 2010.

[27] R. F. Goldin, "Review: Statistical Models-Theory and Practice," *The American Mathematical Monthly,* vol. 117, no. 9, pp. 844-847, 2010.

[28] F. D. Davis, and V. Venkatesh, "Toward Preprototype User Acceptance Testing of New Information Systems: Implications for Software Project Management," *IEEE Transactions on Engineering Management,* vol. 51, no. 1, pp. 31-46, Feb, 2004.

[29] B. Banfai *et al.*, "Implementing an HL7 version 3 modeling tool from an Ecore model," *Studies in health technology and informatics,* vol. 150, pp. 157-61, 2009.

[30] G. Piho *et al.*, "From archetypes-based domain model of clinical laboratory to LIMS software." pp. 1179-1184.

[31] M. Regio, and J. Greenfield, "Designing and Implementing an HL7 Software Factory," 2005.

[32] C. Ohr, and M. Václavík, "Using HL7 Processing Capabilities of the Open Ehealth Integration Platform in the Implementation of IHE Profiles." p. 11, 2009.

[33] R. D. M. Dias, T. W. Cook, and S. M. Freire, "Modeling healthcare authorization and claim submissions using the openEHR dual-model approach," *BMC Medical Informatics and Decision Making,* vol. 11, Oct 12, 2011.

[34] H. Kashfi, and O. Torgersson, "A migration to an openEHR-based clinical application," *Studies in health technology and informatics,* vol. 150, pp. 152-6, 2009.