Maldonado Segura, JA.; Martínez Costa, C.; Moner Cano, D.; Menárguez-Tortosa, M.; Boscá Tomás, D.; Miñarro Giménez, JA.; Fernández-Breis, JT.... (2012). Using the ResearchEHR platform to facilitate the practical application of the EHR standards. Journal of Biomedical Informatics. 45(4):746-762. doi:10.1016/j.jbi.2011.11.004.

Using the ResearchEHR platform to facilitate the practical application of the EHR standards

José Alberto Maldonado[1], Catalina Martínez Costa[2], David Moner[1], Marcos Menárguez-Tortosa[2], Diego Boscá[1], José Antonio Miñarro Giménez[2], Jesualdo Tomás Fernández-Breis[2], Montserrat Robles[1]

[1] *ITACA Institute, Universitat Politécnica de Valéncia, CP 46022, Valencia, Spain*

[2] *Facultad de Informática, Universidad de Murcia, CP 30100, Murcia, Spain*

Email addresses: jamaldo@upv.es (José Alberto Maldonado[1]), cmartinezcosta@um.es (Catalina Martínez Costa[2]), damoca@upv.es (David Moner[1]), marcos@um.es (Marcos Menárguez-Tortosa[2]), diebosto@upv.es (Diego Boscá[1]), jose.minyarro@um.es (José Antonio Miñarro Giménez[2]), jfernand@um.es (Jesualdo Tomás Fernández-Breis[2]), mrobles@upv.es (Montserrat Robles[1])

Preprint submitted to Journal of Biomedical Informatics *October 21, 2011*

Abstract -Possibly the most important requirement to support co-operative work among health professionals and institutions is the ability of sharing EHRs in a meaningful way, and it is widely acknowledged that standardization of data and concepts is a prerequisite to achieve semantic interoperability in any domain. Different international organizations are working on the definition of EHR architectures but the lack of tools that implement them hinders their broad adoption. In this paper we present ResearchEHR, a software platform whose objective is to facilitate the practical application of EHR standards as a way of reaching the desired semantic interoperability. This platform is not only suitable for developing new systems but also for increasing the standardization of existing ones. The work reported here describes how the platform allows for the edition, validation, and search of archetypes, converts legacy data into normalized, archetypes extracts, is able to generate applications from archetypes and finally, transforms archetypes and data extracts into other EHR standards. We also include in this paper how ResearchEHR has made possible the application of the CEN/ISO 13606 standard in a real environment and the lessons learnt with this experience.

Keywords : electronic healthcare records, standards, semantic interoperability, CEN/ISO 13606, archetype, ontology

1. Introduction

Nowadays it is a common scenario that the health data of one patient is scattered among different Electronic Health Record (EHR) systems. This leads to the existence of distributed and heterogeneous data resources creating a large gap between the potential and actual value of the information content of EHR systems. Closing this gap by making efficient use of data held by these systems could improve significantly patient care, patient safety and empower research activities.

Possibly the most important requirement to support co-operative work among health professionals and institutions is the ability of sharing EHRs in a meaningful and secure way. By meaningful we mean that both the sender and receiver system have a common understanding of the content exchanged, i.e. interoperability at the semantic level. Security implies the safe and relevant communication of EHR data whilst respecting the privacy wishes of individual patients. In this context, the focus of this paper will be the meaningful sharing and exchange of EHRs.

It is widely acknowledged that standardization of data and concepts is a prerequisite to achieve semantic interoperability in any domain. This is even more important in the healthcare sector where the need to exchange health data among professional and institutions is not an exception but the rule. The faithful communication of EHRs crucially depends on the standardization of its syntax, structure and semantics, i.e. on the standardization of the EHR architecture and vocabulary used to communicate data. This requires a consistent way for naming and organizing EHR data and concepts in such a way that a requester can precisely specify the desired parts of an EHR and know the data structures that will be provided in response [1]. The challenge lies in finding a generic way to representing every possible EHR structure but at the same time is capable of dealing with the diverse, complex and volatile concepts required by different health care domains.

Different international organizations are or have worked on the definition of an EHR architecture [1, 2]. Health Level 7 (HL7) maintains the XML-based Clinical Document Architecture (CDA) [3] that specifies the structure and semantics of clinical documents for exchange. The Technical Committee 251 (health informatics) of the European Committee for Standardization (CEN/TC251) has completed a European Standard for the communication of the EHR called CEN EN13606 whose part 1 (reference model) [4] became an ISO standard in February 2008 under the name ISO 13606. The openEHR consortium [5] maintains an architecture designed to support the constructions of distributed, patient-centered, life-long, shared care health records. Finally, ISO provides a set of clinical and technical requirements for an EHR architecture that support using, sharing and exchanging EHRs in the technical specification TS 18308:2004 [6].

In spite of the maturity of EHR architectures, result of over fifteen years of research and development, and the recognition of the need to share EHR data between professional and institutions, the set of tools is scarce which hinders their broad adoption. Since EHR architectures define non-trivial models with potentially high nested structures, tooling becomes crucial. Furthermore, as EHR architectures play a central role in EHR communication, in their use we need to cater for different levels of conceptualization potentially ranging from raw data to ontologies.

In the rest of this paper we describe the work carried out in the ResearchEHR project. Its main objective is to provide a software platform for the semantic and standard-based description and sharing of information drawn from legacy EHR systems, supporting healthcare professionals and institutions by providing a set of generic methods and tools for the capture, standardization, integration, description and dissemination of health related information. Semantic interoperability is based on the use of EHR architecture standards, medical terminologies and ontologies, Semantic Web technologies, archetypes and standard to standard semantic transformations. All these technologies are used at different levels, ranging from legacy data all the way to ontological representation of domain concepts. Two different research streams have been carried out. The first one is focused on how to use archetypes to upgrade already deployed systems, in order to make them compatible with an EHR standard. The second one deals with the use of Semantic Web technologies to specify clinical archetypes for advanced EHR architectures and systems. Archetypes play a crucial role in our approach: they represent the meeting point between the semantic-centric modeling and the data-centric modeling of EHRs.

2. Technological background

In this section we present the basic concepts about dual model architectures and ontologies that will make the understanding of our work easier.

2.1. Dual-model architectures

The most remarkable feature of the dual model approach is the separation of information models (such as software models or database schemas), represented by a stable and small information model, from domain models such as blood pressure measurement, discharge report, prescription or microbiology result which are represented by archetypes. Only the stable reference model is hard-coded in database schemas or software, while the possible numerous and volatile domain concepts (archetypes) are modeled separately by domain specialists. Since the software is only bound to the reference model it has no direct dependency on domain concepts. Therefore, systems do not need to be changed when domain concepts are created or altered.

In EHR environments, a reference model represents the generic and stable properties of health record information. It specifies the set of classes that form the generic building blocks of the EHR, how these building blocks should be aggregated to create more complex data structures and the context information that must accompany every piece of data in order to meet ethical, legal and provenance requirements. Although the reference model is standardized across sending and receiving systems it is not enough to describe the full semantics of the domain concepts. The generality of reference models is complemented by the particularity of archetypes. Archetypes are formal definitions of a distinct domain-level concept in the form of constrained combinations of the building blocks defined in the reference model. Their principal purpose is to facilitate the definition of a semantic layer for common understanding and mutual communication of clinical data structured as a set of formal clinical concept definitions decided by health domain experts. The hypothesis behind archetypes is that for each domain concept, a definition can be developed in terms of constraints on structure, types and values of the logical building blocks. The formal description of domain concepts is achieved by linking the data structures and content to knowledge resources such as terminologies and ontologies. Examples of native dual model EHR architectures are CEN/ISO 13606 [4] and openEHR [5].

ADL (Archetype Definition Language) [7] is a formal language developed by openEHR for expressing textually archetypes that has also been adopted by CEN/ISO 13606. An archetype expressed in ADL is composed of four main parts: description, definition, ontology and revision history. The description section basically contains metadata, such as the identifier, the current lifecycle state of development, version, etc. The most important section of an archetype is its definition tree, where the clinical concept is represented by constraining the reference model classes. The ontology section is where the entities specified in the definition section are described and bound to terminologies. Finally the revision history section contains the audit of changes to the archetype. Constraints are written in a block-structured style. The general structure is a recursive writing of constraints on types (known as object nodes or object blocks), followed by constraints on properties of that particular type (known as attribute nodes or attribute blocks), followed once again by constraints of types (being the types of the attribute under which it appears) until leaf nodes (those representing atomic data types) are reached. Names of classes and attributes from the reference model are used for all nodes.

Figure 1 shows an example of a CEN/ISO 13606 blood pressure archetype that describes relevant information related to a blood pressure measurement. The root node of the archetype is an entry (ENTRY[at0000]) which comprises, through the items attribute, a CLUSTER[at0008] grouping the blood pressure information. This cluster contains the following four ELEMENTs under the parts attribute: the systolic measurement (ELEMENT[at0001]), the diastolic measurement (ELEMENT[at0002]), the mean arterial pressure (ELEMENT[at0003]) and the position of the patient while the pressure was measured (ELEMENT[at0004]). The first three elements are physical quantities (a value together with a measurement unit) and the last one a plain or simple text.

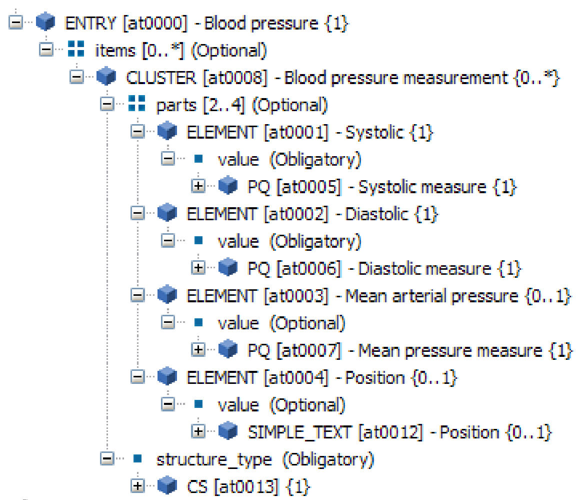

Figure 1: The CEN/ISO 13606 blood pressure archetype

Currently, there exist some initiatives to define more generic and agnostic (in the sense of being independent from reference models) clinical concepts known as Detailed Clinical Models (DCM). DCM have their root in the seminal work of Huff et al. [8, 9]. DCM are similar to archetypes, CDA templates and clinical statements in many ways, see [10] for a detailed discussion. There are several approaches around the world to define DCM, such as the Clinical Elements Model of Intermountain Healthcare [8], the Clinical Contents Model of CiEHR [11], the Logical Record Architecture of the NHS [12] or the Dutch Care Information Model [13]. There are also efforts under way to standardize the definition of DCMs as ISO CD 13972 [14, 15]. Nevertheless the work described in this paper is based on archetypes although some of the tools could be easily extended to support DCM.

2.2. Ontologies

The Semantic Web [16] is a vision of the future Web in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web. There are different basic technologies for the success of the Semantic Web, amongst which the cornerstone technology is the ontology. In the literature, multiple definitions for ontology can be found [17, 18]. An ontology represents a common shareable and reusable view of a particular application domain. It gives meaning to information structures that are exchanged by information systems [19]. The advances in the Semantic Web community make the ontology a candidate technology for supporting knowledge-intensive tasks related to archetypes and EHR systems. Moreover, they have been identified in the final report of the Semantic Health project [20] as one of the basic technologies for the achievement of semantic interoperability of health information systems.

The use of ontologies for representing biomedical knowledge is not new, since they have been widely used in biomedical domains for the last few years with different purposes [21, 22]. In addition to this, recent proposals and approaches support our decision of developing Semantic Web solutions for the management of EHR [23, 24, 25].

These ontologies will be built using the Web Ontology Language (OWL) [26], which is the current W3C recommendation for the exchange of semantic content on the Web. Given that OWL has subspecifications based on Descriptive Logics (DL), OWL ontologies can be exploited by using DL reasoners such as Pellet [27] or Fact++ [28]. This facilitates the development of cost-effective methods for checking the correctness and consistency of medical data, knowledge and archetypes, as we will describe later in this paper.

3. The ResearchEHR platform

The ResearchEHR platform aims at developing and applying semantic technologies for managing existing EHRs. Figure 2 shows the overall architecture of the platform, which can be analyzed from different perspectives.

On the one hand, from an IT perspective, we can analyze it in terms of the level at which the working units are considered, that is, the data level and the ontology level. ResearchEHR includes specific methods and tools for working with both data and ontologies according to the needs of particular tasks. Archetypes play a crucial role in our approach; they represent the meeting point between the semantic-centric modeling and the data-centric modeling of EHRs. The idea behind this is that archetypes facilitate the definition of a semantic layer for common understanding and mutual communication of clinical data. Both works provide interfaces to different worlds: public external information (OWL archetypes) and internal information (already existing EHR sources). The semantic publication of the contents of the archetypes would be in line with the objectives of the development of the Semantic Web, which targets accessible web contents for both humans and computers so that applications might interoperate semantically in an efficient way.

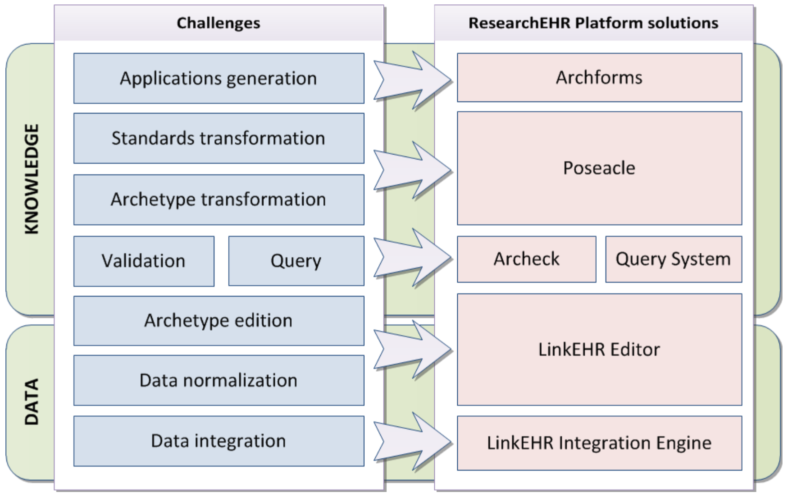

Figure 2: Overall architecture of the ResearchEHR platform

On the other hand, from the EHR perspective, the methods and tools that conform the ResearchEHR platform can be classified according to activities in EHR management: representation, normalization, validation, etc. This will be the approach followed in this paper. Next, we provide an overview of the different activities that ResearchEHR supports:

- Data and Knowledge Representation (see section 4): Archetypes are used to describe the semantics of legacy health data in a manner independent of the particular data organization in the underlying data repositories. This will enable users (mainly health professionals) to view and query data repositories at the level of its relevant semantic concepts. As a main advantage, this approach reduces the problem of knowing the contents and structure of many information sources to the problem of knowing the contents domain-specific concepts (archetypes), which a user familiar with the domain is likely to know or understand easily. On the other hand, archetypes are a mean to achieve semantic interoperability and constitute a basic element in advanced EHR environments. Therefore, these environments will certainly provoke the design of many archetypes for different EHR standards. Hence, mechanisms capable of comparing different archetypes, selecting the best one for a certain purpose, checking the technical correctness and quality of archetypes, and transforming archetypes into different models are required. This issue is approached from a Semantic Web perspective in the ResearchEHR project, since those technologies allow for a better management of clinical information and knowledge.

- Archetype Edition (see section 5.1): A multimodel editor for creating archetypes is included in the ResearchEHR platform. Archetype editing is a process of subtyping by constraints. The rules used to control the subtyping are those specified in the archetype model such as strengthening of domain constraints on primitive attributes or the narrowing of cardinality intervals. These rules are directly implemented in an archetype editor included in the platform and are used by a semantic manager to assist the user in the edition process.

- Archetype Validation (see section 5.2): The current ADL specification is not precise enough regarding archetype semantics, particularly the relationship between reference models and archetypes, which hinders gaining a precise understating of archetypes and their implementation. As a consequence we tackled the task of defining a precise archetype modeling framework as a prerequisite for implementing tools providing enhanced support for archetypes.

- Archetype Search (see section 5.3): The development of archetypes for different standards may convert the process of finding the right archetype in a tedious task. In order to facilitate this task, we have developed an ontology-driven tool for querying an archetype base.

- Archetype-based Applications (see section 5.4): The development of applications based on EHR standards require a deep knowledge of them. In order to facilitate the task of clinical applications developers, ResearchEHR includes a generator of fully working applications from a set of archetypes. This generator is able to produce applications for multiple devices and technologies.

- Legacy Data Normalization (see section 6): Since the health data to be made public resides in the underlying data sources, it is necessary to transform source data to meet the data format of archetypes and reference models. The effort required to create and manage such transformation is considerable. To make this task simpler, ResearchEHR also includes a mapping module between an archetype and a source schema, which is able to generate XQuery programs that transform the source data into a standard-compliant XML documents.

- Interoperability between standards (see section 7): We have developed methods able to transform archetypes and extracts into standards such as CEN/ISO 13606 and openEHR, as well as facilitating the persistence of the archetypes in both ADL and OWL.

4. Archetype Representation

ResearchEHR assumes that there is not a unique representation for archetypes, but that we might need different representations for different challenges. In this way, we are currently managing representations based on the Archetype Object Model (AOM) and OWL.

Archetypes are usually defined in ADL which is a path addressable language that provides an abstract syntax for representing them textually, and uses AOM to express archetypes for any reference model in a standard way. This language has important drawbacks for achieving the goal of semantic interoperability, such as its syntactic orientation. Consequently, the formalization of the exchange and transformation processes is more difficult than using semantic oriented models such as ontological ones. Given this genericity, the language does not provide any component that guarantees the consistency of clinical information. It can only offer consistency at archetype level, that is, the conformance of ADL/AOM principles.

ADL parsers allow reading the ADL archetype and returns a set of generic objects. The reference model data structures and type concepts and their properties, will be represented in AOM by means of C_COMPLEX_OBJECT and C_ATTRIBUTE entities respectively and they will point out to the specific reference model concept or property by means of the string attributes *rmTypeName* and *rmAttributeName* respectively. These objects have no explicit semantic relations between them. Hence, the semantics is unknown for the parser and only the association between elements from the definition and ontology sections might be ideally done by the parser by string matching.

This ADL/AOM representation has some advantages and drawbacks in the context of ResearchEHR. On the positive side, the genericity allows for creating archetypes from different standards easy. On the negative side, we need to develop particular methods for guaranteeing the correctness of the developed archetypes, and this representation is limited to perform semantic activities like comparison or classification, and it is suboptimal for developing automatic methods for processing and exploiting archetypes. Such limitations have been overcome by developing a representation for archetypes based in ontologies.

The ontology layer of ResearchEHR is shown in Figure 3 and comprises the ontologies that model EHR-related knowledge for the different standards. We have developed a series of OWL ontologies for representing the semantics of archetypes [29]. Such ontologies were built from the interpretation of the specification of EHR standards. It should be noted that these ontologies do not model the whole EHR domain as understood in dual modeling architectures, but only the knowledge required to represent archetypes. This means that the ontologies developed cover the archetype model and part of the reference model of the EHR standards. As a result of this interpretation process, three main ontologies were built: (1) the CEN/ISO 13606-RM and OpenEHR-RM ontology, which represents the clinical data structures and data types defined in the reference model of each standards and (2) the ontology which defines the archetype model. The ontologies for CEN/ISO 13606 and openEHR will import the corresponding reference model ontology and the archetype model ontology. In this way, both CEN/ISO 13606 and openEHR ontologies will combine concepts from reference and archetype models linked by means of ontology relationships and will express the archetype structure in a more comprehensible way. The current implementation includes CEN/ISO 13606 and openEHR and the common ontology that will allow in the future, to include other EHR standards in the transformation. The three ontologies share the archetype model but they differ in the reference model definition. The common ontology has been defined to allow representing archetypes from both standards, in this way it includes concepts from both reference models.

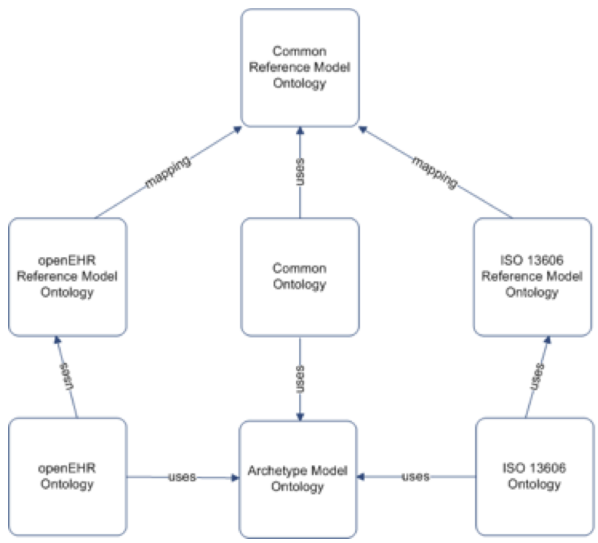

Figure 3: Ontological infrastructure for interoperability

Since archetypes are usually represented in ADL, a methodology for automatically transform them into OWL was designed [30]. This transformation process was implemented by using Model-driven Engineering techniques due to the availability and maturity of tools, and is divided into the three phases shown in Figure 4:

1. The input ADL archetype is expressed as a syntactic model. The tex-

tual ADL archetype is transformed into a model conforming to AOM.

2. This syntactic model is transformed into a semantic model by using a model to model transformation. A set of rules has been defined for the model to model mappings between the syntactic archetype metamodel and the CEN/ISO 13606/openEHR semantic one.
3. The semantic model is transformed into OWL according to the EHR ontologies by using a model to text transformation.

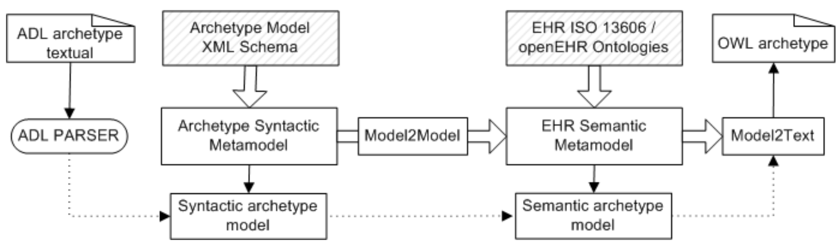

Figure 4: The process for transforming ADL archetypes into OWL

The following modules of the ResearchEHR platform will use the ADL or OWL representation of archetypes depending on which one better fits their needs.

5. Archetype management

5.1. Archetype edition

Our main objective was to develop an archetype editing framework capable of working with several reference models. To the best of our knowledge, the few current archetype editors only support one reference model. In these editors the reference model is hard-coded making very difficult to keep pace with its future evolution or to support other models. Other important issues are how to incorporate a reference model into the tool in run time and how to hide the complexity of reference models to archetype developers. We will briefly discuss these issues in the rest of this section and how they have been implemented in the LinkEHR Editor. For a detailed discussion we refer the reader to [31].

Only a subset of reference model classes can be used to define archetypes, i.e. their specialization can be the root of a domain concept definition. We call them business concepts. For instance, CEN/ISO 13606 defines six explicitly: folder, composition, entry, section, cluster and element. In the case of HL7 CDA, the selection is not so clear. In our applications we have considered those classes with an explicit clinical meaning such as clinicalDocument, Section, Entry and the specializations of clinicalStatement (Observation, RegionOfInterest, SubstanceAdministration, etc).

In the LinkEHR Editor new reference models can be imported at any time as long as they are expressed in a W3C XML Schema. Users need to enumerate the business concepts of the reference model. After this the import module analyses the schemas and as a result it yields a set of archetypes expressed in ADL, one for each business concept, we call them business archetypes. Four reference models have been tested successfully, namely CEN/ISO 13606 [4], openEHR [5], HL7 CDA and CDISC ODM [32]. To the authors' knowledge, LinkEHR Editor is the only archetype tool capable of handling multiple reference models.

The capability of representing reference model concepts as archetypes brings about the possibility of using the same logic for archetype creation either from scratch or by constraining an existing one. Business archetypes are the most general archetypes that can be defined for a reference model and therefore any archetype must be a specialization of one of them. More formally, the archetype editing process becomes a process of subtyping by constraints. As a main consequence, only the specialization rules specified in the archetype model need to be directly implemented in the editor. In the LinkEHR Editor, the semantic validation of an archetype with respect to other archetype (reference model) becomes a matter of finding a subsumption mapping (type assignment) between both archetypes.

Business archetypes are used by a semantic manager to guide users during archetype editing. It guarantees that the archetype being edited is valid with respect the reference model and the parent archetype if exists. At runtime it determines the set of entities (either attributes or types) that are allowed at any point of editing and inform the user. The semantic manager also checks that the constraints on data (cardinality, existence, domain, etc.) are narrower that those specified in the parent archetype.

As stated before archetypes are defined by directly constraining the data structures present in the reference model according to the archetype formalism. This approach forces users to have a deep knowledge of the reference model. In order to make the editor more user-friendly, the editor can incorporate plug-ins. A plug-in defines a customized working environment for a reference model. It contains documentation such as descriptions, on-line tips and hints, term lists and customized visual interfaces that hide the complexity of the structure and non-clinical attributes (mainly attributes holding context data such as dates or identifiers). The current version comes with a plug-in for CEN/ISO 13606 and a new one is being developed for HL7 CDA.

5.2. Semantic validation of archetypes

The detection of inconsistencies in specializations is a major challenge in the process of editing archetypes. Archetypes need to be optimally designed for their purpose, and considered trustworthy within their intended communities of use. In [33], the requirement of formal methods for validating the design and content of archetypes has been identified. An archetype is correct if the set of constraints defined over the reference model and the parent archetype are valid. The specialization of archetypes does not imply inheritance but the definitions in the specialized archetype have to be consistent with the parent's ones.

As it has been previously mentioned, two different representations of archetypes are managed by the platform. Consequently, the semantic validation has to be guaranteed in any of them, which has required the development of different validation methods, which are described next.

5.2.1. Archetype-driven method

Archetypes impose a hierarchical structure to EHR data. For this reason, we employ a data model based on trees, more precisely labeled trees, to formalize the instances of archetypes. Archetypes then become type definition over these labeled trees. The type system uses regular expressions to specify the set of children of a node and label predicates to specify the set of valid labels of a node. Any archetype constraint is modeled either by a regular expression or a label predicate. For instance, regular expressions model existence, occurrence and cardinality constraints, whereas label predicates model domain constraints of primitive types.

We formalize the inheritance relationship between archetypes by means of a subsumption relation [34] based on the containment of regular expression and label predicates. We say that an archetype A is more general than archetype B if A subsumes B. The proposed subsumption relation not only captures the containment relationship between the set of data instances defined by two archetypes but also captures some of the structural relationship between node objects from both archetypes by defining mappings between types. This can be translated to the archetype specialization mechanism. Subsumption mappings specify specialization relationships between the entities (objects and attributes) of the child and parent archetypes. In the editor, we specify reference model classes as archetypes. Therefore, the subsumption relation is also used to formalize the relationship between reference model classes and archetypes: we say that an archetype A specializes a class B if B subsumes A. For a deep discussion on the type system we refer the reader to [31]. LinkEHR Editor uses this approach to validate archetypes that are being edited in the tool.

5.2.2. Ontology-driven method

The combination of advanced semantic models with reasoning techniques reduces the effort required for implementing the quality assurance and validation methods. Our OWL representation of the reference model was achieved by following the rules proposed by the OMG in the Ontology Definition Metamodel specification (ODM) [35].

The semantics of archetype specialization is that the OWL semantics of the parent archetype subsumes the one of the specialized archetype. OWL reasoners allow us to find incorrect constraints over the reference model. Thereby, a concept is wrong defined if the derived OWL class is unsatisfiable.

That is, the set of instances of such concept does not conform to the reference model. OWL reasoners infer subclass and equivalent axioms between classes. In this way, checking the correctness and consistency of a specialization consists on checking whether that subsumption is inferred. Each concept is defined in our representation by means of an OWL class, and its constraints are defined using OWL-DL axioms. Concept identity is associated with the node id, which is used in the archetype definition to bind concepts and ontological definitions. The concepts in specialized archetypes might include additional annotations that guide the validation process. Those annotations indicate the name of the OWL class in the parent archetype that is being specialized, if any. That binding is based on the concept identifier.

An example is shown next. Figure 5 shows the first definitions in Manchester OWL Syntax [36] of the blood pressure archetype (see Figure 1). Each concept is defined in OWL by means of equivalency axioms. The constraints on multivalued associations are also translated into one class. Our OWL representation permits the identification of the classes that violate the definition of the parent archetype. Archeck is the module in charge of performing this validation, as a stand-alone tool but also through a web interface [37].

```
Class: ENTRY_at0000
       EquivalentTo: ENTRY and ARCHETYPED_CLASS and (id value "at0000")
                                    and (op_items only COLLECTION_ENTRY_at0000_items)

Class: COLLECTION_ENTRY_at0000_items
       EquivalentTo: COLLECTION and (id value "COLLECTION_ENTRY_at0000_items")
                                    and (elements only CLUSTER_at0008)

Class: CLUSTER_at0008
       EquivalentTo: CLUSTER and ARCHETYPED_CLASS and (id value "at0008")
                                    and (op_parts only COLLECTION_CLUSTER_at0008_parts)

Class: COLLECTION_CLUSTER_at0008_parts
       EquivalentTo: COLLECTION and (id value "COLLECTION_CLUSTER_at0008_parts")
                                    and (elements only (ELEMENT_at0001 or ELEMENT_at0002 or
                                         ELEMENT_at0003 or ELEMENT_at0004))
                                    and (elements min 2 ITEM) and (elements max 4 ITEM)
                                    and (elements exactly 1 ELEMENT_at0001)
                                    and (elements exactly 1 ELEMENT_at0002)
                                    and (elements max 1 ELEMENT_at0003)
                                    and (elements max 1 ELEMENT_at0004)
...
```

Figure 5: Excerpt of the blood pressure archetype in OWL

5.3. Ontology-driven archetype querying

The advanced semantic query subsystem aims to define queries driven by the ontologies described in Section 4. It provides a graphical interface that suggests at each step which elements can be included in the query to avoid inconsistencies whereas allowing a higher level of expressiveness than traditional querying interfaces.

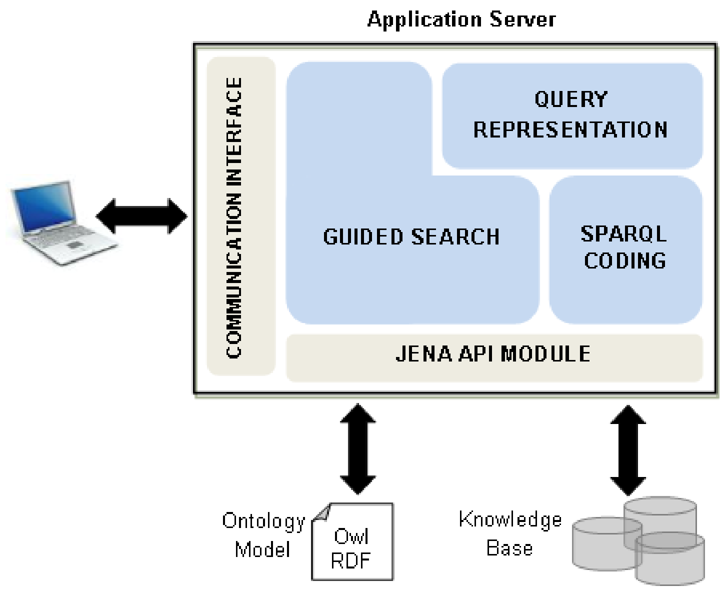

Figure 6: The architecture of the advanced semantic querying system

Figure 6 shows its architecture, which consists of three submodules and two interfaces. The module *Guided Search* is responsible for providing proper ontological resources to users according to the *Archetype Ontology* and the status of the definition of the query. So, this module guarantees that there are no syntactic errors or inconsistencies in the definition of the query by limiting the options that users can make (see Figure 7).

The module *Query Representation* stores the status of the query and its information during the definition process in order to assist the module Guided Search. The module *SPARQL Coding Module* is in charge of transforming the query definition into a well formed SPARQL query, and querying the *Archetype Base*. On the other hand, the *Communication Interface* shows graphically the options that the module Guided Search make available to the users during the definition process. Finally, the interface *Jena API Module* is used to handle the RDF/OWL documents and to perform the SPARQL queries in the Archetype Base.

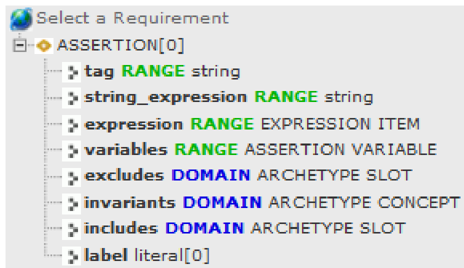

Figure 7: An example of the graphical interface where the choices are provided to users to define the query requirements

This subsystem reduces the complexity of knowing semantic query languages, such as SPARQL, and the URIs used in the underlying ontology of the archetype base. However, the users still need to be conscious of the ontology and its structure to know how to relate the terms to define the proper query. More information about this system can be found in [38].

5.4. From archetypes to applications

ArchForms [39] is the generator of applications from archetypes included in ResearchEHR. The functionality of the generated applications are not only be data input, but also data validation and generation of data extracts compliant with a particular dual model-based EHR standard. The resulting web applications generate CEN/ISO 13606 compliant EHR extracts in XML format to exchange data with other systems and to facilitate the insertion of data in the already existing databases. The approach used for developing this web application generator does not only allow the generation of applications for a particular technology, but it can be easily adapted to different software platforms, user interfaces and devices. This is possible because the process for generating the application transforms first the archetype into a generic representation of the graphical interface based on XForms [40], and further stages take both the archetype and that graphical interface to generate a full application.

The generative architecture of ArchForms has two main phases: (1) generation of the generic GUI models; and (2) generation of the code for the target platform. The input of the generation process are the ontology representation of an archetype obtained from an ADL archetype. In order to generate the application forms, archetypes are represented as generic GUI models, independently from a particular interface technology. A set of XForms models are generated from each ENTRY concept of the archetype, independently from the specific user interface implementation technology. Once the generic GUI models are available, the source code of the application for a particular platform can be generated. First, one web form is generated for each generic GUI model obtained in phase one. Second, the classes that support the validation of the data input by the users and the persistence of EHR extracts.

The current generator includes three different implementation technologies: Seam, TouchFaces and PrimeFaces. The generated applications share the functionality but they differ in how the user interacts with the particular device. Special attention has been paid to the usability and compatibility of the web applications based on TouchFaces. These applications have been tested for Android and iPhone. Figure 8 shows the corresponding application example for iPhone for the blood pressure archetype (see Figure 1).

Figure 8: Blood pressure iPhone application

6. Integration and standardization of legacy EHR data

6.1. Data access and integration

The LinkEHR Integration Engine (LinkEHR-IE) is the module that gives access to existing data at the original health information systems. LinkEHR-IE can be classified as a generic middleware that integrates clinical information available in distributed and heterogeneous data sources [41]. It allows the definition and management of a global, integrated and structured XML view of all the clinical records stored for a patient that is generated on demand [42].

LinkEHR-IE is based on Integration Message Definitions (IMD). They describe the clinical concepts that can be shared among the different subsystems involved in an integration project. Each IMD can only be shared as a whole. In other words, the minimum unit of information that can be shared between two subsystems is generated by an IMD entity. An IMD definition includes the specification of the data sources to be used, the data items (databases, tables and fields) to be extracted from each data source, the input parameters accepted by query processor to execute filters on data sources and the definition of elements that describe the labeling and nesting format that constitutes the resulting XML document.

Once we have defined the needed IMDs for a use case, LinkEHR-IE can be deployed. When an information request is received, the appropriate IMD definitions are loaded and executed, thus querying the original data sources in order to retrieve legacy data and integrating them into a single unified XML view. Additionally, a transformation can be applied to the integrated data in order to normalize it (see section 6.2) or to make it readable (a typical XML-to-HTML XSLT transformation). The result can be returned to the requester or directly rendered in a viewer.

6.2. Generation of standardized EHR extracts from legacy data

One of the main problems when adopting EHR-related standards is the standardization of existing data. This problem is a difficult one, since it deals with differences and mismatches between heterogeneous formats and models. In our scenario, this problem is even more complex. On one side, we have the legacy data that conform to a particular schema and with local semantics. On the other side, we have EHR architectures and archetypes that intend to be as generic as possible. Therefore, they are defined without any consideration regarding the internal architecture or database design of EHR systems. Our objective is to create an instance of the target schema (archetype) taking data structured under the source schema (legacy EHR). For this purposes, we require an explicit representation of how the source schema and target schema are related to each other. These explicit representations are called (schema) mappings. To be useful in practice, mappings should be "compiled" into an executable implementation, for example, under the form of SQL queries for relational data, XSLT or XQuery scripts for XML data. The target instances should possess a set of desirable properties, such as they must be legal instances for the target schema, they contain all the source information and at the same time they are non-redundant [43, 44]. All this involves several challenges related to the semantics of schema mappings and the generation of code based on these mappings.

As stated before, the potential mismatch regarding structure and semantics between legacy EHR data and archetypes is big. Therefore, complex and expressive mappings between them are required. Our first approach to map EHR data and archetypes was to use available commercial tools. These tools are capable of handling XML schemas, but since archetypes cannot be expressed as XML Schemas these tools are not suitable for our purpose.

In ADL only the constrained entities (classes and attributes) of the reference model need to appear in the archetype definition. This rule poses a difficulty when an archetype is mapped to a data source. In many cases it would be necessary to map an unconstrained attribute, hence not present in the archetype. Note that our final objective is to generate XML documents compliant with the reference model. Thus, when an archetype needs to be mapped it becomes necessary to complete the archetype definition with the reference model. We have implemented a merge function that takes an archetype and the underlying reference model as inputs and outputs what we call a comprehensive archetype. A comprehensive archetype includes all the explicit constraints (those defined by the archetype to be mapped) and all the implicit ones (those defined by the reference model) that data instances must satisfy. Figure 9 shows an example of comprehensive archetype. On the left-hand side the original CEN/ISO 13606 archetype is depicted, whereas the corresponding comprehensive archetype is shown on the right-hand side. As it can be observed the comprehensive archetype contains all the constraints of the original archetype as well as all the unconstrained entities from the reference model such as act_status, archetype_id, etc.

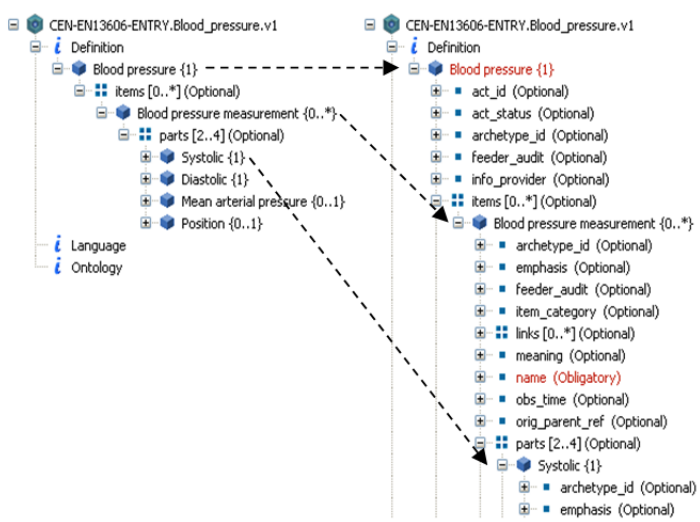

Figure 9: The blood pressure comprehensive archetype

In LinkEHR we have followed the common approach for mapping systems. Users are responsible of defining a high-level non-procedural mapping representation that does not cover all the mapping details. Then, based on this representation, the tool generates the actual data translation program (low-level mapping) by working out the missing details. Figure 10 shows the overall architecture of our archetype mapping systems. In our case, the specification is defined by a set of correspondences between entities of archetypes and source schemas (either an XML Schema or other archetype). Two types of correspondences are supported: between atomic entities (leaf nodes) and between complex entities (inner nodes). The former are value correspondences that specify how to calculate atomic values whereas the latter are structural correspondences that may be used to control the generation and grouping of elements in the target.

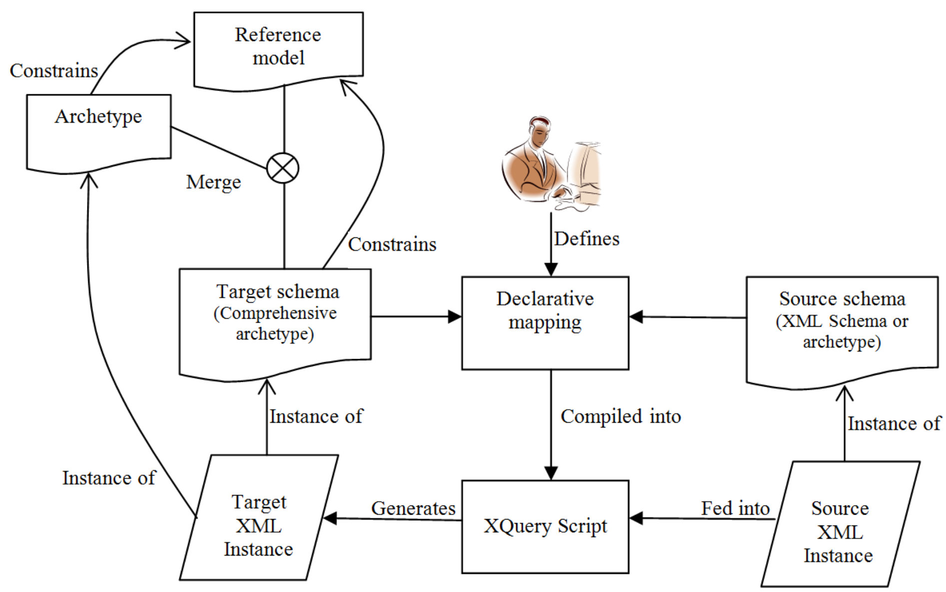

Figure 10: The archetype mapping and XQuery generation process

Value correspondences are defined by an ordered set of pairs. Each pair contains a transformation function and a filter. The transformation function specifies how to calculate a value in the target from a set of source values.

The simplest kind of transformation function is the identity function which copies a source value or a constant into a target value, but we can also make use of a set of complex functions that have been defined into the tool, such as type conversion, mathematical, logical, string, date and time functions. Regarding the filter, it stipulates the conditions that source data must satisfy in order to be used in the transformation function. The default filter is the "true" predicate. According to the specified order only the first applicable function is used. For instance table 1 contains a simple value correspondence. It should be interpret as: if /patient/value is lower or equal to 0, then return 0; else if /patient/value is between 0 and 20, then return /patient/value*2; else return 40.

Filter	Function
/patient/value<=0	0
/patient/value>0 AND /patient/value<=20	/patient/value *2
true	40

Table 1: Sample mapping table

EHR reference models and archetypes may define complex structures with depth nesting that makes the mapping specification a very complex task. A first key requirement is represented by ease of use, in the sense that this complexity should be hidden as much as possible. Value correspondences are easy to specify. For instance, users do not have to fully specify the logical relations (e.g. parent-child relations) between the entities of the schemas. It is only necessary to specify the navigation path of the involved attributes. But value correspondences lack expressive power and some mapping details must be worked out [45]. A key aspect is the grouping semantics, i.e. when target instances must be grouped and nested inside the same element. LinkEHR

Editor comes with a default grouping semantics based on Partition Normal Form. This default grouping strategy has resulted to be adequate in many cases. Basically due to the fact that is context-aware, in the sense that data with the same clinical context are grouped together. For instance, in the case of CEN/ISO 13606 data that share the same committed time, attester, etc. are grouped together.

The default semantics depends on the structure of the target schema (archetype). In some occasion it is necessary to take into account the structure of the source schema [46]. Structural correspondences are used for this purpose. Structural correspondences are defined by a set of source paths and a filter. They control the creation of target instances, in such a way that a new target instance is constructed for each set of source nodes addressed by the paths that satisfies the filter.

Taking into account the abstract mapping specification, the archetype constraints and the source schema an XQuery script is generated. The script takes as input an instance of the source EHR data and generates a XML document that is compliant both with the archetype and the underlying reference model.

7. Interoperability between standards

ResearchEHR provides a methodology to enable EHR systems based on dual model architecture but using different reference models to exchange clinical data. It consists of two transformation steps: (1) archetype transformation and (2) data transformation. In the first one, archetypes used for capturing data in a specific system are transformed into valid archetypes for other standard. In the second transformation step, data captured and already standardized in one of the systems are transformed into valid data for other standard. The current implementation permits the transformation between CEN/ISO 13606 and openEHR and has been implemented as a web service and as a web application named Poseacle Converter [47].

7.1. Archetype transformation

As mentioned in section 4 we proposed ontologies for representing clinical information semantics. However, for the technical transformation of archetypes, we represent them as models because of the maturity of tools and availability of languages for doing the transformations. Therefore, in the architecture proposed, two layers are distinguished (see Fig 11): (1) ontology layer and (2) metamodel layer.

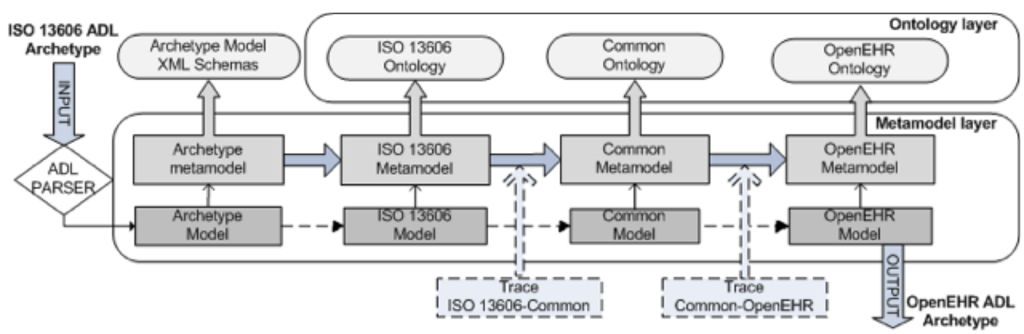

Figure 11: Archetype transformation process

The ontology layer has been described in section 4. The metamodel layer contains the metamodels corresponding to the semantic representations defined in the ontology layer. Consequently, metamodels for the CEN/ISO 13606, openEHR and the Common ontologies were developed by using the ODM standard. Once the metamodels have been obtained, the correspondences among them were defined. The mappings used can be found in [48]. They have been defined between the particular standard and the Common metamodel and implemented using the model transformation language RubyTL. This language allows defining a set of transformation rules that establish the correspondences between objects of the metamodels.

The archetypes are then transformed by using the following workflow:

- The ADL archetype input is transformed into its MDE representation conforming to the Archetype metamodel (Archetype model).
- The Archetype model is transformed into the Source EHR representation (source model)
- The Source model is transformed into the common archetype representation (Common model).
- The Common model is transformed into Target representation (target model).
- The Target model is transformed into ADL code (target ADL archetype).

The RubyTL language allows obtaining a model with the trace of the transformation. This model allows knowing the transformation mappings that have been applied and it will be very useful in order to perform the data transformation. As it can be observed in Figure 11, two trace models are obtained as a result of each archetype transformation. They will be processed in order to obtain an only model that includes the mappings applied between ISO 13606 and openEHR and it is named semantic trace.

7.2. Standardized data transformation

In dual model-based EHR standards, data is usually represented as XML extracts. Therefore, in order to transform data extracts, the mappings established between archetypes have to be applied to data. Figure 12 depicts the data transformation process, which require the following steps:

- Generation of the syntactic mappings: Each piece of data in an XML extract is identified by a syntactic path, which is the path of a concept or property in the extract. The set of pairs of syntactic paths that define the mappings between two standards are named syntactic mappings. The syntactic paths for both standard representations can be obtained from the ADL archetypes. In order to define the syntactic mappings, the correspondences established at archetype level are used, that is the semantic trace mentioned before.

- Transformation of data: The syntactic mappings will be used to access data in the source standard representation and define them according the target standard.

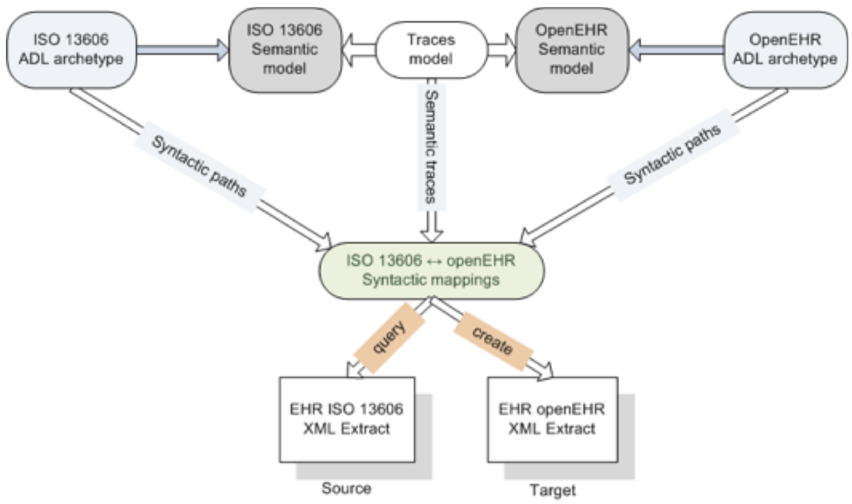

Figure 12: Data transformation process

8. Validation in a real setting

The main goal of this research was to create a platform that could help in the utilization of EHR standards for the description and communication of health data, thus supporting faithful EHR sharing. The use case, described below, was aimed at evaluating this goal. The evaluation study was the development of a software platform for medicines reconciliation. Medicines reconciliation is the process of obtaining and evaluating an up-to-date and complete medication list in order to avoid medication errors such as omissions, duplications, dosing errors or drug interactions. It should be done at every transition of care in which new medications are ordered or existing orders are altered. In this project, we cooperated with the Hospital of Fuenlabrada (Spain) in a project whose objective was the development of tools to guarantee the reconciliation between the hospital and the primary care centers of its health area.

The technical solution was based on the use of the CEN/ISO 13606 norm for the description and sharing of medication-related data. We chose CEN/ISO 13606 for two main reasons: i) it defines a generic and simple reference model that allows rapid development and is understood by health and IT professionals easily, ii) it is a "native" dual model and therefore was defined to be used along with archetypes. Four data sources were used: the primary care information system, the hospital electronic health record system, the pharmacy information system of the hospital and the Spanish National Medication Database (Nomenclator Digitalis). The concepts to be shared were modeled as archetypes. For the definition of archetypes we took into account three main sources: the Patient Summary specifications developed by the epSOS European project [49], several archetypes of the openEHR Clinical Knowledge Manager (CKM) [50] and the NEHTA specifications for medications [51]. The Patient Summary archetype was directly created with the LinkEHR Editor. Regarding the openEHR archetypes, they were transformed into their representation according to the CEN/ISO 13606 by using the Poseacle Converter tool and their correctness were checked by using the Archeck validation tool.

The epSOS initiative is the main European electronic Health interoperability project co-funded by the European Commission. Its main focus is the improvement of medical treatment of citizens while abroad by providing health professionals with the necessary patient data. Since the medication information is part of the epSOS patient summary, we decided to use the epSOS patient summary as the container data set of the medication data. The main advantage was that with the same effort and technological infrastructure the hospital was ready to share normalized patient summaries. Figure 13 depicts the archetype used to describe the medication information at both primary care and the hospital information systems.

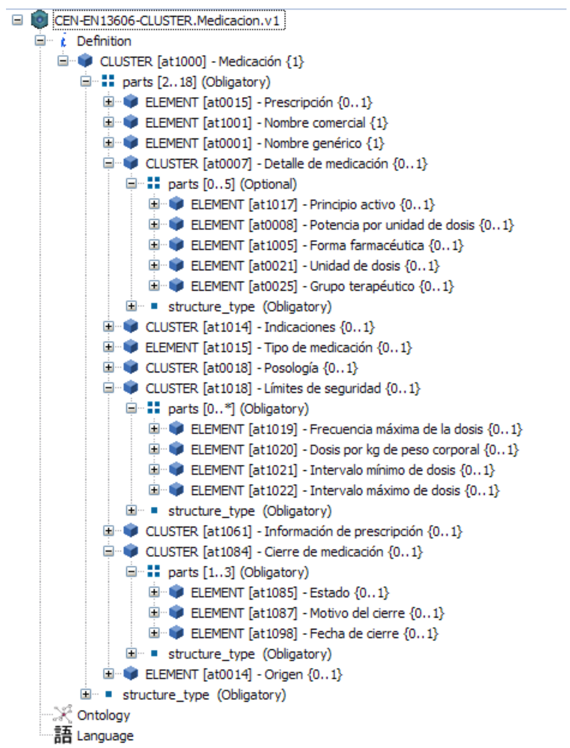

Figure 13: CEN/ISO 13606 medication archetype

The archetypes were validated by the clinical team composed by the Medical Director of the hospital, the head of the Pharmacy Service and other technical and clinical staff. The next step was to use the mapping capabilities of the LinkEHR Editor in order to map the data sources schemas to the archetype. Once again, the collaboration of the hospital members, both medical and technical, was a key success factor. They helped in identifying the location and meaning of each data item at their data base systems and they validated the final results. The mapping lead to the generation of an XQuery script that once deployed at the original systems and applied to existing data, is able to generate a standardized view or XML extract of existing and not normalized data. Figure 14 depicts an example of clinical data represented according to the patient summary archetype that describes partly the medication according to the archetype shown before. The root node of the patient summary archetype is a composition (COMPOSITION[at0000]) which comprises the ENTRY[at0003]. This entry defines a medication table (CLUSTER[at0004]) that contains the following two ELEMENTs: the drug brand name (ELEMENT[at1001]) and its source (ELEMENT[at0014]). Their values will be described by using coded or simple text data types. It defines the brand name of the drug as plain text (*RESOURCE DIABET 24 COMBIBLOCS 200ML*) together with the Spanish National Medication Database code (340430) and its origin (Primary Health Care setting - *Atención primaria* in Spanish).

```xml
<EHR_EXTRACT>
  <all_compositions type="COMPOSITION">
    <name type="SIMPLE_TEXT">
      <originalText>Historia cl\'inica resumida-Paciente: MDSC299</originalText>
    </name>
    <archetype_id>CEN-EN13606-COMPOSITION.HCR_Fuenlabrada.v1</archetype_id>
    <content type="ENTRY">
      <name type="SIMPLE_TEXT">
        <originalText>Medicaciones</originalText>
      </name>
      <archetype_id>at0003</archetype_id>
      <items type="CLUSTER">
        <name type="SIMPLE_TEXT">
          <originalText>Tabla de medicaciones</originalText>
        </name>
        <archetype_id>at0004</archetype_id>
        <parts type="CLUSTER">
          <name type="SIMPLE_TEXT">
            <originalText>Medicación</originalText>
          </name>
          <archetype_id>at1000</archetype_id>
          <parts type="ELEMENT">
            <name type="SIMPLE\_TEXT">
              <originalText>Nombre comercial</originalText>
            </name>
            <archetype_id>at0101</archetype_id>
            <value type="CODED_TEXT">
              <archetype_id>at0002</archetype_id>
              <originalText>RESOURCE DIABET 24 COMBIBLOCS 200ML</originalText>
              <codedValue type="CD">
                <codingSchemeName>Nomenclator</codingSchemeName>
                <codeValue>340430</codeValue>
              </codedValue>
            </value>
          </parts>
          <parts type="ELEMENT">
            <name type="SIMPLE_TEXT">
              <originalText>Origen</originalText>
            </name>
            <archetype_id>at0014</archetype_id>
            <value type="SIMPLE\_TEXT">
              <archetype_id>at0031</archetype_id>
              <originalText>Atenci\'on Primaria</originalText>
            </value>
          </parts>
        </parts>
      </items>
    </content>
  </all_compositions>
</EHR_EXTRACT>
```

Figure 14: Excerpt of an CEN/ISO 13606 patient summary extract that describes a medication prescription

The hospital EHR viewer was upgraded to include a new tab for the patient summary. As a result when a physician is reviewing a patient EHR, they have an access to the patient summary that includes the medication list generated from data coming from both the hospital and the primary care centers. Then they can act properly when prescribing medications assigned to the patient to avoid safety risks.

Currently, over 430 physicians and 600 nurses from both the hospital and primary care centers are using the system to gain access to a normalized patient summary of more than 230.000 people. The key for the success of the project was the deep involvement of the medical director and the clinical staff, since they were responsible of defining the clinical content of the archetypes and the logical correspondences to existing databases.

In addition to the definition of CEN/ISO 13606 archetypes for the description and sharing of medication-related data, the Poseacle Converter tool was used in order to get also their openEHR representation. In this way, it is also possible to transform the clinical data defined according to CEN/ISO 13606 into valid clinical data conforming to openEHR. Figure 15 depicts the resulting openEHR extract for the running example. There it can be observed how the different data structures and types have been transformed into the corresponding openEHR ones like GENERIC_ENTRY, DV_TEXT or DV_CODED_TEXT.

```
<composition type="COMPOSITION">
  <archetype_node_id>openEHR-RM-COMPOSITION.HCR_Fuenlabrada.v1</archetype_node_id>
  <content type="GENERIC\_ENTRY">
    <archetype_node_id>at0003</archetype_node_id>
    <data type="ITEM_TREE">
      <items type="CLUSTER">
        <archetype_node_id>at0004</archetype_node_id>
        <name type="DV_TEXT">
          <value>Tabla de medicaciones</value>
        </name>
        <items type="CLUSTER">
          <archetype_node_id>at1000</archetype_node_id>
          <name type="DV_TEXT">
            <value>Medicación</value>
          </name>
          <items type="ELEMENT">
            <name type="DV_TEXT">
              <value>Nombre comercial</value>
            </name>
            <archetype_node_id>at0101</archetype_node_id>
            <value type="DV_CODED_TEXT">
              <archetype_node_id>at0002</archetype_node_id>
              <value>RESOURCE DIABET 24 COMBIBLOCS 200ML</value>
              <defining_code type="CODE_PHRASE">
                <terminology_id>Nomenclator</terminology_id>
                <code_string>340430</code_string>
              </defining_code>
            </value>
          </items>
          <items type="ELEMENT">
            <name type="DV_TEXT">
              <value>Origen</value>
            </name>
            <archetype_node_id>at0014</archetype_node_id>
            <value type="DV_TEXT">
              <archetype_node_id>at0031</archetype_node_id>
              <value>Atención primaria</value>
            </value>
          </items>
        </items>
      </items>
  </data>
  </content>
</composition>
```

Figure 15: Excerpt of the openEHR patient summary extract that describes a medication prescription

9. Discussion

One of the key factors for the successful deployment of standards is to seamlessly integrate the new developments within the existing information systems. The ResearchEHR platform was designed and developed with this idea in mind, enforcing the reuse of data and knowledge available at health organizations. This approach can be clearly seen at the use case of medications reconciliation that has been described.

First, we must be able to construct a complete view of the existing data related to the patient and the use case. The LinkEHR Integration Engine helps in building such view by querying and integrating data from heterogeneous and distributed data sources on demand. At the medicines reconciliation case, this module has been used for extracting clinical data (allergies, problems and medications) from the different data sources, mainly the primary care information system and the hospital EHR system. The data integration field has been active for many years. We can find many commercial and free solutions dealing with it such as Mirth, InterSystems Ensemble or Orion Rhapsody, but they are focused mainly toward the integration of applications and messages rather than integration of data. LinkEHR-IE is a lightweight system that simplifies this process providing a flexible mechanism for defining and generating integrated data extracts from data bases or existing documents. From our perspective, dealing with messages, business logics and systems integrations is a task to be performed at a higher level, once the data is already normalized in a standard model.

The second step is to normalize those data into a standard representation. The LinkEHR Editor provides the two needed functionalities at this point. On the one hand, it can be used by clinical users to define formal concepts or archetypes, following any standard or proprietary reference model. On the other hand, it is a powerful tool for mapping and normalization of non-standardized data based on those same archetypes. There exist tools for both defining archetypes and mappings between schemas, but LinkEHR Editor is the first one designed to perform both tasks natively, i.e. it supports both the editing of archetypes and the definition of mappings between legacy data and archetypes or between archetypes. For example, another archetype editor is the openEHR Archetype Editor. It can be used to create archetypes but only supports the openEHR reference model. Moreover, it has no functionality for binding those archetypes to existing data. As an example of a data transformation tool, we can find Altova Mapforce. It is a powerful data binding and transformation tool but it does not support the concept of archetypes. Data transformed with LinkEHR Editor has not only a different format, but in fact is semantically enriched through the used archetype definitions. This difference is also demonstrated at the use case, where we have been able to merge information from heterogeneous sources into a homogeneous and standardized view from a semantic point of view. It is not only about putting several data together in the same format, but to assure that those data are semantically equivalent and can be described by the same concepts (archetypes). This has been especially true while working with medications from primary care and the hospital. At primary care, medications are prescribed by their commercial or brand name. At the hospital the prescriptions are made by active ingredient, dose form and strength. The defined medication archetype covers both methods, so that it can accommodate data from the two different systems. Moreover, due to the use of terminologies such as the Spanish National Medications Database, we have been able to automatically transform any commercial reference or brand name to its equivalent active ingredient+dose form+strength, to ease the understanding of this information by the hospital professionals. Thus, the use of clinical terminologies, together with archetypes, has demonstrated to be a fundamental aspect for semantic enrichment of existing data.

The third step is to deal with several standards. We can define our own archetypes or use archetypes that have been previously defined. For example, we can find hundreds of openEHR archetypes at their online knowledge repository [50]. A tool to allow the reuse of all those definitions was also needed, and that was provided by the Poseacle Converter. This tool is able of generating both ADL/OWL representations of archetypes from CEN/ISO 13606 and openEHR and XML data extracts for such standards. Its interoperability infrastructure has been carefully designed to be able to include other standards in the future and, according to our knowledge, it is the first implementation of such transformations.

At the fourth step, we have to check the validity of archetypes: the existing ones, the automatically transformed ones and the newly created ones. LinkEHR Editor and Archeck are two different approaches to solve this problem and the need for both of them is drawn from the nature of the ResearchEHR platform. We aim at developing activities that deal with both data and knowledge in which archetypes are the common entities to both types of activities. As it has been previously mentioned, there is not a unique representation of archetypes and this is what ResearchEHR reflects.

LinkEHR Editor has an incorporated mechanism to assure that archetypes that are being edited are completely valid regarding the underlying reference model. Since this edition is guided by the archetype model itself, it is impossible to add any property, class or attribute that does not exist at the reference model. Moreover, it has an incorporated validation algorithm that checks that any archetype is subsumed by its parent or reference model archetype. Given that this process is guided by the XML schemas of the EHR standard, transforming this into the ontological space would result in a real-time additional time greater than implementing the validation at this level. On the other hand, the interoperability and semantic activities included in ResearchEHR are based on a semantic representation of the archetypes. Given that archetypes are mainly written and distributed in ADL, semantic activities require the transformation of ADL archetypes into valid semantic representations. In addition to this, it could be possible to receive archetypes generated in OWL from other systems. In such cases, the AOL-driven validation would not be enough since. Archeck would then validate those archetypes in such cases. This modules also demonstrate that the OWL-DL representation of archetypes and the application of DL reasoning may save time and cost in the development of solutions for the semantic validation of archetypes.

We also use all those archetypes and semantic definitions for advanced uses, such as the querying and the generation of applications which have not been applied in this use case. Archforms is the generator of applications included in ResearchEHR, providing a simple way of generating applications from archetypes. This module was not used in the use case because one of the requirements of the hospital was the adaptation of its current system and the goal of the use case was the visualization of information rather than the input of new data, which is the aim of Archforms applications. However, in future similar use cases we would like to study whether the adaptation of Archforms applications to existing systems is more cost and time-effective than the development of new modules for existing applications.

The only suite that could be comparable to ResearchEHR is the set of tools developed in the the openEHR world, but there are some basic differences between both suites. On the one hand, ResearchEHR has been developed with a multimodel orientation. Despite some tools might be available so far only for CEN/ISO 13606, the transformation tools between standards could be used to fulfill this limitation. On the contrary, openEHR tools are only available for the openEHR model. In addition to this, the openEHR tools cannot deal with ontological archetypes and activities as ResearchEHR does. However, the current version of ResearchEHR does not have an archetype manager like the one developed by the openEHR Foundation. Our future plans include to incorporate into the platform ArchMS [52], our prototype for multi-model archetype management and we are also working in a standard-independent detailed clinical model (DCM) definition framework. The availability of such an archetype repository would also facilitate the usage of the advanced query subsystem, which currently requires a specific configuration of an archetype base. This system has the advantage of facilitating the design of semantic queries in which the archetype ontologies drive the process. In order to design a query, the users describe the properties they are looking for in the archetype base and the queries are issued by taking into account the formalization included in the ontology such as the definition of transitive or symmetric properties.

10. Conclusions

Traditionally, computer systems process data without worrying of their meaning. But there is an increasing need of taking into account the meaning of that data in order to meaningfully share it, to better understand it and to generate knowledge from it. The ResearchEHR platform we have introduced represents a step ahead toward semantic interoperability of health information systems. Based on the use of standards and archetypes we can deliver a set of tools and services that cover the needs for semantic enrichment and interoperability of existing data. A stack of innovative technologies has been developed and integrated including the access and integration of data, the normalization of existing information structures, the semantic modelling of concepts through archetypes, the reuse of those concepts among different standards and the exploitation of those standardized data. This software platform represents a valuable resource towards a better use of health information. As the medications reconciliation has showed, even with the most advanced tools, the help and collaboration of clinical users is unavoidable. They are the ones with the exact knowledge about what they want, need and wish. Software developers cannot replace them or take decisions only on a technical basis. The experiences gained by the authors during the development of the platform show that archetypes are a suitable mechanism to improve the communication between both worlds. They represent a formal definition of the domain knowledge described by the final users themselves and a design contract for software and tools developers.

Finally, once the barrier of a good representation of information and concepts has been broken, a new world of possibilities is opened. We have showed some of the possibilities, such as applications generation and querying, but the possibilities go beyond. One of the most promising uses includes the improvement of clinical research methods, by enabling a seamlessly access to information from multiple patients in a standardized way. We can also think of advanced decision support systems or the automatic integration of electronic health record systems and personal health record systems. Any of these will require to work over a formal basis regarding data standardization and semantic description of those data, and the ResearchEHR platform is aimed to help on this duty.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Science and Innovation under Grant TIN2010-21388-C02-01 and TIN2010-21388-C02-02, and by the Health Institute Carlos III through the RETICS Combiomed, RD07/0067/2001. Our most sincere thanks to the Hospital of Fuenlabrada in Madrid, including its Medical Director Pablo Serrano together with Marta Terrón and Luis Lechuga for their support and work during the development of the medications reconciliation project.

References

[1] D. Kalra, Electronic health record standards., Yearbook of medical informatics (2006) 136–44.

[2] M. Eichelberg, T. Aden, J. Riesmeier, A. Dogac, G. B. Laleci, A survey and analysis of electronic healthcare record standards, ACM Comput. Surv. 37 (2005) 277–315.

[3] Health Level 7 International, Clinical Document Architecture, http://www.hl7.org/implement/standards/cda.cfm, Last accessed: July 2011.

[4] ISO – International Organization for standardization, Health informatics – Electronic health record communication – Part 1: Reference model, 2008.

[5] openEHR consortium, `http://www.openehr.org`, Last accessed: July 2011.

[6] International Organization for Standardization, Requirements for an electronic health record architecture, Technical Report, Technical Specification TS 18308, Last accessed: July 2011.

[7] openEHR consortium, Archetype Definition Language ADL2, `http://www.openehr.org/releases/1.0.2/architecture/am/adl2.pdf.`, Last accessed: July 2011.

[8] S. Huff, R. Rocha, J. Coyle, S. Narus, Integrating detailed clinical models into application development tools, Stud Health Technol Inform 107 (2004) 1058–62.

[9] C. Parker, R. Rocha, J. Campbell, S. Tu, S. Huff, Detailed clinical models for sharable, executable guidelines, Stud Health Technol Inform 107 (2004) 145–8.

[10] W. Goossen, A. Goossen-Baremans, M. van der Zel, Detailed clinical models: A review, Healthcare Informatics Research 16(4) (2010) 201–14.

[11] Center for Interoperable EHR, Clinical contents model, http://ehrkorea.org/, Last accessed: July 2011.

[12] National Health Service UK, Logical Record Architecture for Health and Social Care, http://www.connectingforhealth.nhs.uk/systemsandservices/data/lra, Last accessed: July 2011.

[13] J. van der Kooij, W. Goossen, A. Goossen-Baremans, N. Plaisier, Evaluation of documents that integrate knowledge, terminology and information models, Stud Health Technol Inform 122 (2006) 519–22.

[14] ISO – International Organization for standardization, NWIP/CD ISO 13972-1 Health Informatics: Detailed Clinical Models Part 1. Quality processes regarding detailed clinical model development, governance, publishing and maintenance, 2011.

[15] ISO – International Organization for standardization, NWIP ISO 13972-2 Health Informatics: Detailed Clinical Models Part 2: Quality attributes of detailed clinical models, 2011.

[16] T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web, Scientific American 284 (2001) 34–43.

[17] T. R. Gruber, A translation approach to portable ontology specifications, Knowledge Acquisition. Special issue: Current issues in knowledge 5(2) (1993) 199–220.

[18] G. van Heijst, A. Schreiber, B. Wielinga, Using explicit ontologies in kbs development, Int. J. Hum.-Comput. Stud. 46 (1997) 183–292.

[19] C. Brewster, K. O'Hara, S. Fuller, Y. Wilks, E. Franconi, M. A. Musen, J. Ellman, S. B. Shum, Knowledge representation with ontologies: The present and future, IEEE Intelligent Systems 19 (2004) 72–81.

[20] V. Stroetman, D. Kalra, P. Lewalle, A. Rector, J. Rodrigues, K. Stroetman, G. Surjan, B. Ustun, M. Virtanen, P. Zanstra, Semantic Interoperability for Better health and Safer Healthcare, Technical Report ISBN-13: 978-92-79-11139-6, European Commission, 2009.

[21] S. Schulz, U. Hahn, Part-whole representation and reasoning in formal biomedical ontologies 34 (2005) 179–200.

[22] B. Smith, From concepts to clinical reality: An essay on the benchmarking of biomedical terminologies, Journal of Biomedical Informatics 39 (2006) 288–98.

[23] J. Grabenweger, G. Duftschmid, Ontologies and their application in electronic health records, in: eHealth 2008 Medical Informatics meets eHealth, Wien.

[24] B. Blobel, C. Gonzalez, F. Oemig, D. M. Lopez, P. Nykänen, P. Ruotsalainen, The role of architecture and ontology for interoperability, in: B. Blobel, E. Hvannberg, V. Gunnarsdottir (Eds.), Seamless Care - Safe Care: The Challenges of Interoperability and Patient Safety in Health Care, volume 155, IOS Press, Amsterdam u.a., 2010, pp. 33–9.

[25] C. Tao, G. Jiang, W. Wei, H. Solbrig, C. Chute, Towards Semantic-Web Based Representation and Harmonization of StandardMeta-data Models for Clinical Studies, in: Amia Annual Symposium, 2011, ISBN 0-9647743-9-9.

[26] OWL Web Ontology Language Reference, http://www.w3.org/tr/owl-ref/, Last accessed: July 2011.

[27] Pellet OWL reasoner, http://pellet.owldl.com/, Last accessed: July 2011.

[28] Fact++ OWL reasoner, http://owl.man.ac.uk/factplusplus/, Last accessed: July 2011.

[29] J. T. Fernández-Breis, M. Menárguez-Tortosa, D. Moner, R. Valencia-García, J. A. Maldonado, P. J. Vivancos-Vicente, T. Miranda-Mena, R. Martínez-Béjar, An ontological infrastructure for the semantic integration of clinical archetypes, Lecture Notes in Computer Science 4303 (2006) 156–67.

[30] C. Martínez-Costa, M. Menárguez-Tortosa, J. Fernández-Breis, J. Maldonado, A model-driven approach for representing clinical archetypes for semantic web environments, Journal of Biomedical Informatics 42 (2009) 150–64.

[31] J. A. Maldonado, D. Moner, D. Bosca, J. T. Fernández-Breis, C. Angulo, M. Robles, LinkEHR-Ed: A multi-reference model archetype editor based on formal semantics, International Journal of Medical Informatics 78(8) (2009) 559–70.

[32] CDISC, ODM - Operational Data Model, http://www.cdisc.org/odm, Last accessed: July 2011.

[33] D. Kalra, EHR archetypes in practice: getting feedback from clinicians and the role of EuroRec, in: EuroRec - EHTEL Conference, Vienna. eHealth Planning and Management Symposium.

[34] G. M. Kuper, Subsumption for xml types, in: In Proceedings of International Conference on Database Theory (ICDT), Springer LNCS, 2001, pp. 331–45.

[35] Ontology Definition Metamodel specification v1.0, http://www.omg.org/docs/formal/09-05-01.pdf, Last accessed: July 2011.

[36] W3C: Owl 2 web ontology language manchester syntax, http://www.w3.org/TR/owl2-manchester-syntax/, Last accessed: July 2011.

[37] M. Menarguez-Tortosa, C. Martinez-Costa, J. Fernandez-Breis, Validating Archetypes using OWL Reasoning, in: II OpenHealth Spain Symposium, Fuenlabrada, Madrid, 2010.

[38] C. Martínez-Costa, J. A. Miñarro Giménez, M. Menárguez-Tortosa, R. Valencia-García, J. T. Fernández-Breis, Flexible semantic querying of clinical archetypes, in: Proceedings of the 14th international conference on Knowledge-based and intelligent information and engineering systems: Part I, KES'10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 597–606.

[39] C. Martinez-Costa, M. Menarguez-Tortosa, J. Fernandez-Breis, Archforms: Generación automática de aplicaciones iso 13606, in: Las TIC en

la sostenibilidad de los Sistemas para la Salud y el Bienestar. Inforsalud 2010.

[40] XForms, `http://www.w3.org/TR/xforms11/`, Last accessed: July 2011.

[41] G. Wiederhold, Mediators in the architecture of future information systems, IEEE Computer 25 (1992) 38–49.

[42] C. Angulo, P. Crespo, J. A. Maldonado, D. Moner, D. Pérez, I. Abad, J. Mandingorra, M. Robles, Non-invasive light-weight integration engine for building ehr from autonomous distributed systems., International Journal of Medical Informatics 76(S3) (2008) 417–24.

[43] R. Fagin, P. G. Kolaitis, L. Popa, Data exchange: getting to the core, ACM Trans. Database Syst. 30 (2005) 174–210.

[44] A. Bonifati, G. Mecca, P. Papotti, Y. Velegrakis, Discovery and correctness of schema mapping transformations, in: Schema Matching and Mapping, Springer Berlin Heidelberg, 2011, pp. 111–47.

[45] L. Popa, Y. Velegrakis, M. A. Hernández, R. J. Miller, R. Fagin, Translating web data, in: Proceedings of the 28th international conference on Very Large Data Bases, VLDB '02, VLDB Endowment, 2002, pp. 598–609.

[46] A. Raffio, D. Braga, S. Ceri, P. Papotti, M. A. Hernandez, Clip: a visual language for explicit schema mappings, in: Proceedings of the 2008 IEEE 24th International Conference on Data Engineering, IEEE Computer Society, Washington, DC, USA, 2008, pp. 30–9.

[47] Poseacle converter, http://miuras.inf.um.es:9080/PoseacleConverter, Last accessed:July 2011.

[48] C. Martínez-Costa, M. Menárguez-Tortosa, J. T. Fernández-Breis, An approach for the semantic interoperability of ISO EN 13606 and OpenEHR archetypes, Journal of Biomedical Informatics 43 (2010) 736–46.

[49] European Patients Smart Open Services - epSOS, http://www.epsos.eu, Last accessed: July 2011.

[50] openEHR consortium, Clinical Knowledge Manager - CKM, http://www.openehr.org/knowledge/, Last accessed: July 2011.

[51] NEHTA, Medication Data Specifications, http://www.nehta.gov.au/, Last accessed: July 2011.

[52] J. Fernández-Breis, M. Menárguez-Tortosa, C. Martínez-Costa, E. Fernández-Breis, J. Herrero-Sempere, D. M. Cano, J. Sánchez-Cuadrado, R. Valencia-García, M. Robles-Viejo, A Semantic Web-based System for Managing Clinical Archetypes, in: Conf Proc IEEE Eng Med Biol Soc., pp. 1482–5.