A Conceptual Data Model for Health Information Systems

André Magno Costa de Araújo[1], **Valéria Cesário Times**[1], **Sérgio Castelo Branco Soares**[1]
[1] Center for Informatics, Federal University of Pernambuco, Recife, Pernambuco, Brazil

Abstract - *The development of Health Information Systems based on dual models allows modifications to be conducted in the layer of archetypes, reducing dependencies on software developers. However, we identified a lack of conceptual models to represent two-level database entities. This paper proposes a novel conceptual data model, called ArcheER, which is a dual modeling approach and aims to reduce redundant entities and guarantee the creation of unique electronic health records. ArcheER is an extension of the Entity-Relationship model and is based on archetypes. A CASE modeling tool based on ArcheER is outlined. Finally, to illustrate the key features of the proposed model, an ArcheER conceptual schema built for a legacy system is discussed, and results collected from a test with 18 human subjects are reported. Results indicated a reduction of 83,35% in the representation of redundant entities and a gain of 78,9% concerning the modeling of entities characterizing knowledge.*

Keywords: Novel Software Tools, Conceptual Data modeling, Health Information Systems, Archetypes.

1 Introduction

Conceptual modeling is an important activity for designing a database. The conceptual scheme is a concise description of data requirements specified by the application designer, including detailed descriptions about types of entities, relationships and constraints [1]. Thus, the artifacts generated from the conceptual data modeling are important elements in building database systems. Currently, most Health Information Systems (HIS) are built using traditional database modeling technologies [2], in which both information and knowledge concepts are represented in single level computer systems using conventional data models. However, HIS must handle a large number of concepts that often change or are specialized after a short period of time and, consequently, HISs based on such models are expensive to maintain.

Several research projects and many applications have been developed from the specifications of the openEHR system architecture and the concept of archetypes [3-8]. The Open Electronic Health Record (openEHR) software architecture for HIS is aimed at developing an open, interoperable and computational platform for the Health domain [9]. This architecture separates generic information that represents the structures of the Electronic Health Records (EHR) and demographic characteristics of the patients of a reference model, from the constraints and standards associated with the clinical data of a given specific domain, which composes the knowledge model. An archetype consists of a computational expression that is based on the reference model and is represented by domain constraints and terminologies [3] (e.g. data attributes of a blood test), while templates are structures used to group archetypes for allowing their use in a particular context of application, and are often associated with a graphical user interface. On the other hand, some authors have already proposed extensions of traditional conceptual modeling techniques to represent HIS applications. However, these extensions do not model EHR, do not provide dual modeling constructors and are not based on archetypes. In fact, little attention has been devoted to the investigation of the following issue: which conceptual constructors are needed to model the two-level database entities of HIS applications?

This paper proposes a novel conceptual two-level data model, named ArcheER, for helping database designers with the modeling of HIS applications. ArcheER is an extension of the Entity-Relationship (ER) model [1] and is based on the openEHR definitions [3]. It also comprises a set of modeling constructors with graphical representations for building health information conceptual schemas and a set of knowledge-level constructors that are based on archetypes. The ER model was chosen because it is simple and widely used in both academia an organizations for the development of DB applications, and because it is capable of providing an abstraction of implementation details as well as being easily mapped to DBMS logical data models. Another contribution of this paper is related to the development of a modeling tool for ArcheER. The main goal of such tool is to provide application designers with computer support to assist in the database modeling activities of healthcare applications.

The dual modeling approach has been used by several researchers and is not unique to archetypes [10]. However, in this paper the focus lies on the use of dual modeling based on the concept of archetypes, since [8],[11],[12] reported the use of this approach as essential to achieve interoperability and standardization of EHR.

2 Related work and motivation

Späth and Grimson [8] used the openEHR specification to map the structure of an EHR into a proprietary database system. They examined the reuse of archetypes available in the repository of openEHR by specializing some of them and then proposing a new set of archetypes to support biomedical knowledge discovery. To achieve this, they studied the database schema to reorganize it according to the concept of archetypes, by mapping each field of the database to an archetyped element. Some difficulties were reported while doing so, including a lack of consolidated modeling tools and lack of mechanisms to determine overlapping archetypes as well as solve the semantic conflicts that may appear when archetypes are mapped to the chosen DBMS.

Bernstein et. al [6] conducted a study in Denmark about the patterns of the development of healthcare computer systems. This research indicated that the Danish healthcare systems were based on several information models and heterogeneous technology platforms, developed by different software vendors. Besides, it showed the need for replacing traditional standards of software development in the Health domain, and reported the importance of the openEHR architecture as a new pattern for the development of computer systems for healthcare.

Despite the development of HIS based on the openEHR specifications being a multidisciplinary research area, (there are already varied studies published by the scientific community [13-15]) there is a consensus that the openEHR architecture definitions have to evolve and address some open problems [8]. This paper points out that the difficulty in applying the openEHR concepts to a given problem domain for enabling the two-level data modeling is due to the lack of a methodology to express which are the data requirements requested by users and how these might be modeled.

This paper goes one step beyond previous works by specifying an ER-based conceptual data model enabling the definition of which archetypes, patient demographic properties, hospital administrative information and clinical data are important and should be taken into account during the conceptual modeling of a healthcare database application. The main concepts of the ArcheER modeling proposal are detailed in the following section.

3 The ArcheER conceptual model

ArcheER is a conceptual data model that aims to allow the specification of a health application domain using the concepts of dual modeling. The proposed set of ArcheER modeling constructors extends the basic ER modeling elements with a set of archetyped components listed in Figure 1. ArcheER represents entity types alongside their relationships and properties, and a conceptual schema is composed of hospital administrative data and archetyped information. An archetyped entity denotes a set of entities that must have a set of generic data structures. Each of those structures is defined as an attribute of those entities, and organizes data through data structuring elements that are neither dependent of the DBMS storage format nor of the application development technology.

In order to model relationships between archetyped entities, and a relationship between a conventional entity and an archetyped entity, ArcheER proposes a new relationship type, called *Party Relationship*. It is worth noting that an administrative entity may not be modeled as an archetyped entity, i.e. it may not be represented as an entity type with a set of generic data structures, therefore not being able to have party relationship associations with other entity types. However, this must not be mandatory and an administrative entity may benefit from the use of generic data structures as well.

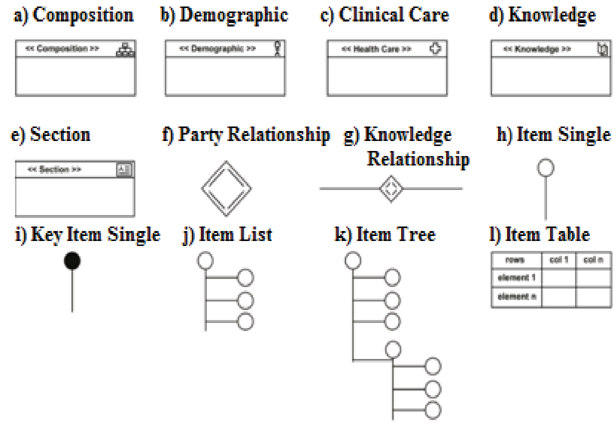

Figure 1. The Main Modeling Components of ArcheER

One of the advantages of ArcheER is the elimination of data redundancy by defining a uniqueness constraint based on the concept of roles played by the actors being modeled. According to this constraint, every instance of a relationship involving the demographic information of an actor and an entity of the type *clinical care* or *administrative* must be modeled as a relationship between a role played by the actor and the entity type *clinical care* or *administrative*. Observe that the use of openEHR definitions requires an understanding of which actors should be considered while modeling an application domain, how they relate to each other, how they play their roles and which capabilities they have. This understanding is important and must not be neglected. However, this cannot be enforced automatically by the DBMS nor can it be seen as a data model constraint to guarantee a unique EHR.

The ArcheER constructors inherited from ER are mostly used for modeling the operational aspects of a hospital organization (i.e. entity type *Administrative*), while the archetyped entity types are mainly concerned with the representation of (i) metadata and the context of the application being modeled (i.e. entity type *Structuring*); (ii) patient's demographic information (i.e. entity type *Demographic*.); (iii) clinical data (i.e. entity type *Clinical Care*) and (iv) constraints, terminologies of health area, internal coding of vocabulary and textual information given by a domain specialist (i.e. entity type *Knowledge*). While the first three entity types represent the information level of the dual modeling approach, the last type of entity and its specializations compose the second level and are useful for generating knowledge at runtime. The definition of each type of constructor is given below.

3.1 Structuring constructors

The ArcheER data model provides the following modeling constructors for structuring health care information: *Composition*, whose attributes represent the metadata of an ArcheER conceptual schema; and *Section,* which organizes the remaining modeling constructors of ArcheER into themes or subjects that represent the context of the application being modeled.

3.2 Demographic constructors

The modeling of demographic information requires the identification of actors who compose the hospital application domain, and the definition of their roles and capabilities in the health area. For modeling actors, ArcheER specifies the following set of constructors of demographic entities that represent the specialization of an actor in a Health domain. The definition of each type of demographic entity is given below:

- *Agent*: Expresses a software agent or any device that communicates with the healthcare application.
- *Person*: Corresponds to an entity type that represents a generic description of people who are part of the context of the application being modeled.
- *Group*: Models parts of the real world that interact with each other and are grouped to represent the purpose of being together.
- *Organization*: Denotes an abstraction of all companies involved in a health application domain.
- *Role*: Represents a generic description of a role played by a given actor.
- *Capability*: Models the qualification of an actor to play a certain role in a healthcare domain.
- *Party Identity*: Indicates how an actor is identified in a healthcare application, and allows an actor to be identified in several ways.
- *Contact*: Expresses the possible ways of contacting an actor.
- *Address*: Indicates how the contact information of actors is formatted.

3.3 Health care constructors

The ArcheER modeling constructors that represent health care information are in charge of defining all the semantics of EHR - hence, the information they model represent the main target to be archetyped. For the modeling of clinical care information, ArcheER proposes the following entity types:

- *Admin Entry*: Expresses all the administrative information of patients in the modeling of EHR. Note that this entity type concerns the modeling requirements of the patient´s administrative information that compose the EHR of the patient and does not refer to administrative aspects of a service provider organization in health. In this work, for the modeling of these administrative issues of a health service organization, we assume that the use of traditional ER constructors will suffice, thus, in fact, only clinical care and demographic information are modeled using archetypes.
- *Observation*: Represents any event or clinical status associated with the patient.
- *Instruction*: Expresses all future actions to be administered to the patient.
- *Activity*: Specifies the activities of an instruction.
- *Action*: Specifies the actions of an instruction.
- *Evaluation*: Represents general information about the clinical care of patients, based on diagnosis, assumptions, risk assessments and observations.

3.4 Knowledge constructors

The entity type *Knowledge* expresses the terminology and constraints related to attributes, also called generic data structures. This entity allows the second level of dual modeling to be displayed in an ArcheER schema. Furthermore, ArcheER adds a new constructor of relationships, called *Knowledge Relationship,* to express associations between generic data structures and instances of the entity type *Knowledge.*

ArcheER extends the definition of ER relationship types to enable the creation of direct relationships between the generic structure attributes of archetyped entities and the entity type *Knowledge*. The following relationship cardinalities are considered by our ArcheER proposal: 1:1, 1: N and M: N.

The entity type *Knowledge* is specialized in the following entity types: *Free Text, Internal Code and Terminology*, which are directly related to the generic data structures through the *Knowledge*-type relationship. These specializations model knowledge of a given Health domain – in other words, they represent the second level of ArcheER dual modeling. The entity type *Free Text* represents free text information given by the domain specialist, while the entity type *Internal Code* denotes codes of a health vocabulary (e.g. procedures, billing tables, international classification of diseases) used for the exchange of information between EHR applications. Lastly, the entity type *Terminology* represents terms and concepts designed to standardize, promote and disseminate health knowledge.

3.5 Data entry constructors

The ArcheER modeling constructors used to define attributes are called data entry constructors, since such attributes comprehend entries with any kind of data that are represented by generic data structures. Hence, generic data structures are defined as attributes of archetyped entities of ArcheER. For each element of these data structures, a data type must be specified. An entry may have a single clinical statement (e.g. a short description about the history of the current illness), or otherwise contain a large amount of data (e.g. the list of values of a laboratory test, tabular data reporting a hospital infection, a hierarchical structure containing all procedures, materials and medications of a patient's hospital bill, an entire microbiology result or a psychiatric examination note). An entry defines the semantics of multiple formats of data which are properties of the archetyped entities of ArcheER.

For modeling generic data structures, the ArcheER model provides the following types of attributes: *ITEM_SINGLE*: represents a data structure with a single element; *ITEM_LIST*: represents a list of data items or values, where each element of this list may assume a value or not, may be referenced by a name and may have an index to indicate its position within the list; *ITEM_TREE*: models a data structure that is logically represented as a tree; and *ITEM_TABLE*: defines a data structure with lines and columns, where the line represents the specification of an element, and the column the information value.

3.6 ArcheER constraints

A set of constraints aiming at ensuring the uniqueness of the EHR is specified in Object Constraint Language (OCL) [16] notation. Thus, the relationship between demographic and clinical care entities and the relationship between demographic and administrative entities of the patient are restricted by two constraints, respectively: *Context Clinical Care inv: Health->forAll (oclType=Role) and Context AdminEntry inv: Health->forAll (oclType=Role or OclType=Administrative)*. Consequently, each instance of entities *CareEntry* and *AdminEntry* is related with demographic information of patients only by means of the roles played by the actors. The benefit of defining constraints over these relationships using the concept of roles is that, while actors of a Health domain are modeled as generic entities, their specific characteristics are represented as roles. This ensures the conceptual modeling of the uniqueness of demographic information, since new instances of a given actor are created only through the roles played by him.

In order to model actors, roles and capabilities, we propose four constraints, which are aimed at enforcing the uniqueness of EHR and explained as follows. Constraint *Context Actor inv: Actor.allInstances->forAll (ar | self.Actor < > ar.Actor implies self < > ar)* specifies the actors' uniqueness constraint and enforces that each instance of an actor entity of ArcheER is unique. The constraint *Context Actor inv: self.Actor_Role->notEmpty() implies self.Actor_Role->forAll (r1 | self.Role < > r1.Role implies self < > r1)* indicates that each actor is not allowed to have two instances with the same role, while constraint *Context Role inv: self.Actor_Role-> includes (self.Actor)* guarantees that in order to create a new instance of a given role, a corresponding instance of an actor must exist. In addition, the constraint *Context Capability inv: self.Role_Cap-> includes (self.Role)* defines that, for each instance of the entity *Capability*, a corresponding instance of the entity *Role* must exist. Entity *Address* models the details of each instance of the entity *Contact*; thus, an instance of entity *Address* can exist only if there is a corresponding instance of entity *Contact*. This is enforced by the constraint *Context Address inv: self.address-> includes (self.Contact)*.

The entity *Administrative* may be related to demographic information (i.e. to instances of the entity *Demographic*) and to clinical concepts as well (i.e. to instances of the entity *ClinicalCare*). For relationships with demographic information, a constraint is specified to ensure that this relationship is always established through instances of the entity *Role*, while for the relationship with a clinical care entity, there must be an entity *AdminEntry*. The constraint *Context Administrative inv: demographic-> forAll (oclType=Role or oclType=AdminEntry)* guarantees this.

3.7 The ArcheER case tool

ArcheERCASE is a computational modeling tool that builds conceptual data schemas based on ArcheER. It is a graphic design software, not a technology-oriented tool, since both the data schema elaborated using this tool and the configuration metadata of this tool are stored in XML format.

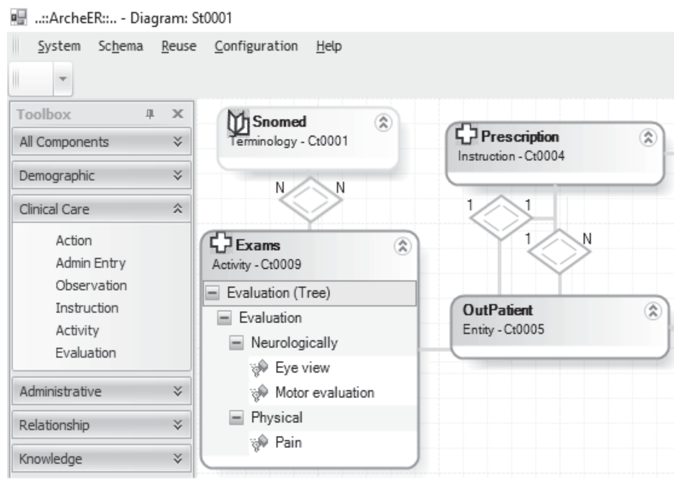

Figure 2. The ArcheERCASE Tool

The main goal of ArcheERCASE is to provide application designers with computer support to assist in the database modeling activities of healthcare applications. Details about ArcheERCASE, including the system prototype architecture, the ArcheERCASE Data Dictionary and the Graphic Module of ArcheERCASE can be found at www.r2asistemas.com.br/ArcheER. Figure 2 depicts the graphic environment of this tool alongside graphic notations.

4 Results

4.1 Experimental design

To validate ArcheER, we conducted two data modeling experiments with two distinct set of human subjects. In both experiments, nine Brazilian professionals with at least two-year experience in conceptual modeling and database design were asked to build two conceptual schemas to model a problem domain. The experiment is based on a hospital scenario located in Northern Brazil, for urgency care. The full description of the problem domain is available at www.r2asistemas.com.br/archeER.

The goal of this research is not to determine the best conceptual data model for HIS applications, but to better understand some important differences between ArcheER and ER conceptual models.

To accomplish that, we computed the time each participant took to complete a given modeling task. Also, for each conceptual data schema generated, we observed whether the uniqueness of EHR was represented, and whether terminologies used in health standards were identified and modeled. Observing the software artifact produced by the ArcheER approach (i.e. each ArcheER conceptual scheme), our experiments measure the difference with respect to the ER model in the following aspects: (i) elapsed time for building a conceptual data scheme; (ii) number of redundant entities produced by each conceptual data schema; and (iii) number of entities that represent terminologies and standards in health of each conceptual data schema produced.

To perform our experiments, each selected participant received the following support instruments: a) instruction about the ER model, b) instruction about the ArcheER approach, c) record sheet, and d) description of the problem domain. In our experiments, the following hypotheses were considered: Hypothesis 1 (H1): The use of the ArcheER approach reduced the time needed to build a conceptual data scheme of a problem domain. Hypothesis 2 (H2): The use of ArcheER warrants the uniqueness of EHR. Hypothesis 3 (H3): ArcheER allows the identification of terminologies and health standards used in a given problem domain.

The variables considered in our experiments are: F1 – Conceptual Modeling Technique for building data schemes; Level of factor T1: Conceptual scheme designed with the ER model (F1→T1); Level of factor T2: Conceptual scheme designed with the ArcheER model (F1→T2). The metrics collected in our experiments are TSB – Time spent for building the conceptual data scheme, QRE – Quantity of redundant entities and QEK – Quantity of entities characterizing knowledge (terminologies and standards). The subjects selected to take part in this study were divided into two working groups (i.e. G1 and G2), chosen by lottery. To eliminate the influence of previous experience of the selected subjects, we used the design of Latin Square experiment 2x2. Considering that Exp1 and Exp2 correspond to the experimental objects that were randomly attributed by lottery to the variables, the experiment design is described in Figure 3a.

Design		
	Exp1	Exp2
G1	F→T1	F→T2
G2	F→T2	F→T1

a) Latin Square

	Group 1			Group 2		
	TSB	QRE	QEK	TSB	QRE	QEK
Stat t	4.73	12.57	10.66	0.40	7.44	4.35
t-student	1.79	1.81	1.75	1.76	1.83	1.81

b) Statistical results of each metric

Figure 3. Design and statistical results

For interpreting the raised hypothesis, we have used the *t* distribution test. This test is often chosen when the average population is less than 30 and there is a normal (or approximately normal) distribution. For this work, the distribution of *t* sampling with n-1 degrees of freedom was adopted. Figure 3b has the values of each metric computed after the application of statistical tests.

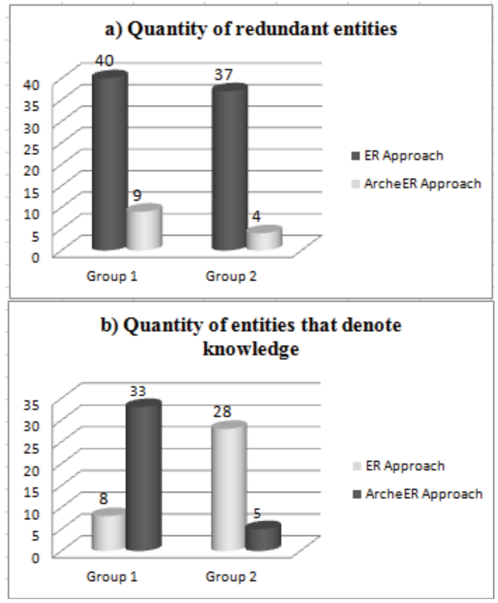

Figure 4. Results of experimental design

Results indicated that the time for building a conceptual data scheme is similar for both modeling approaches. However, for the other two metrics, it is possible to say that, as shown in Figure 4a, the quantity of redundant entities in conceptual schemes designed by the ER Model is greater than the respective number of redundant entities in schemes designed using ArcheER. Moreover, the quantity of redundant entities was reduced in 77.5 % for group 1 and in 89.2 % for group 2 with the adoption of ArcheER approach. Actually, in conventional modeling, for each new role an actor plays in a health domain, new instances are created to represent it, which possibly generates data redundancy in the DBMS – i.e., if a doctor needs to be represented as a patient, a new instance of patient is created by storing information about this person redundantly in the EHR.

Regarding the quantity of entities that denote knowledge, the ArcheER approach identified more entities than the ER approach, as shown in Figure 4b. This increase represents a gain of 75.7% for group1 and 82.1% for group 2. As the use of health terminologies and standards is common in the Health domain, the previous identification of terminologies and standards during the conceptual modeling phase can provide a better understanding of which archetypes are needed for an application to generate knowledge during runtime.

4.2 Modeling an outpatient emergency with ArcheER

In this section, we describe the main difficulties encountered in modeling HIS using traditional approaches, and later we comment on the advantages of modeling HIS using ArcheER. For the sake of didactics, we present in Figure 5 a data schema extracted from a HIS produced by manufacturers of a Health Software in Brazil. This HIS concerns an ambulatory emergency that is performed daily at a Hospital located in Northern Brazil.

Observing the data schema, it is possible to see that the initial difficulty is due to the variety of roles played by the actors in a Health domain, such as workers of a hospital, physicians responsible for patient care, nurses, and other health professionals that sometimes act as health care providers, but occasionally might be seen as a patient who receives care themselves. Besides, the current approaches for database modeling do not provide any constraints to limit this redundancy. Actually, in conventional modeling, for each role played by an actor in a Health domain, new instances are created to represent it, and thus data redundancy may be added to the DBMS.

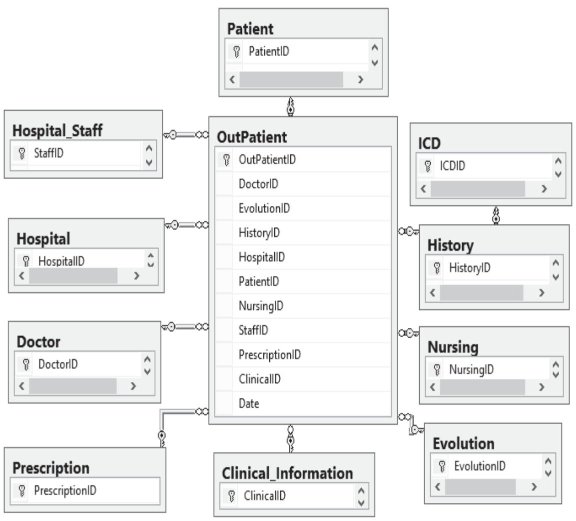

Figure 5. Legacy Data Schema

It is possible to see, in Figure 5, that entities representing demographic information (i.e. Doctor, Hospital, Hospital_Staff, Patient and Nursing_Staff) reflect this modeling practice. In other words, if an actor plays a role, new instances are created for each entity, making their information redundant in the EHR.

In the ArcheER model proposal, actors are modeled in their more generic way, with new instances being created from the roles played. Therefore, an actor may have several roles in an organization and still keep its record unique. As shown in Figure 6, the entity *Person_EHR* represents the most generic characteristics of the actor, while entities *Hospital_Staff, Patient, Nursing_Staff* and *Doctor* represent the roles played by this actor in EHR. To play a role, the actor must have training that qualifies them to perform the referred role – in this case, the *Council* entity illustrated in Figure 6 represents the professional record that the actor needs to have in order to play the role of a physician.

Besides the roles played in a Health domain, an actor may take the form of an organization that provides health services, or that is directly involved in the application context. In this sense, the entity *Hospital* of Figure 6 represents the organization responsible for providing services to the patient. This case shows that the advantages of the ArcheER model go beyond the input of demographic information into the EHR modeling: due to the specified constraints, a demographic entity may only be related to other concepts of EHR (i.e. clinical care, administrative) by means of a role played. In this case, if necessary, only new instances of the roles played by an actor are created, keeping its most generic characteristics preserved, thus ensuring the uniqueness of EHR. As Figure 6 shows, all relationships having an entity that represents patient care (i.e. OutPatient) are established by means of the roles identified in the described application.

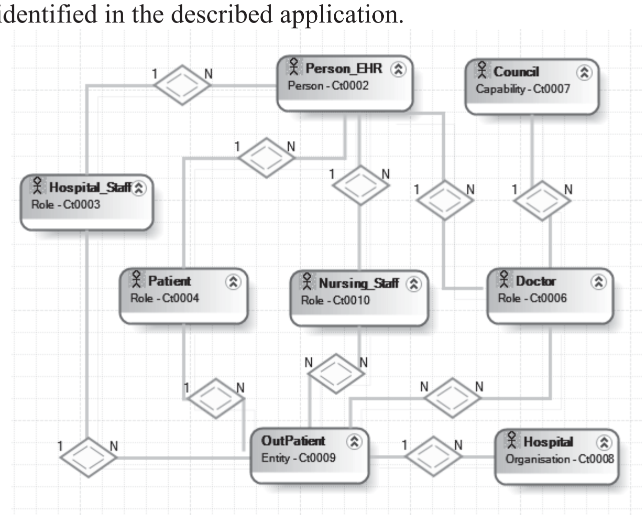

Figure 6. Demographic Conceptual Schema

Figure 7 portrays entities that model clinical care, administrative and knowledge information. Entities *Snomed*, *List_Presc* and *ICD* show the knowledge modeled in the ArcheER conceptual schema. The first entity expresses the terminology and constraints of health care regarding the construction of laboratory examinations, while the entity *Item_Presc* models an internal coding that standardizes the prescription items of a hospital. Finally, the entity *ICD* represents the terminology used to define the patient diagnosis internationally.

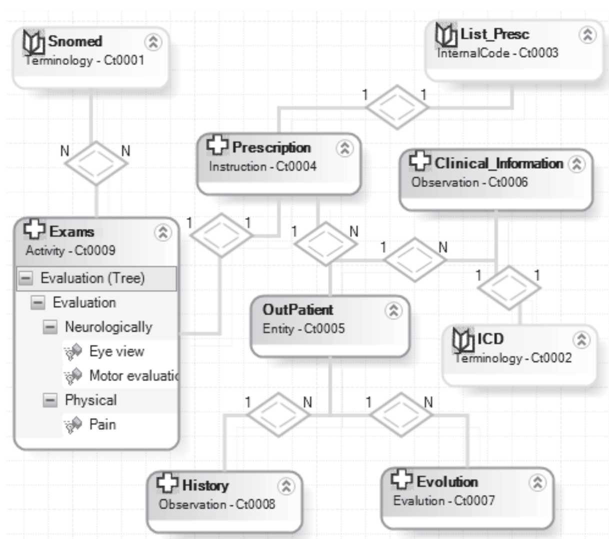

Fig. 7.Clinical Conceptual Schema

For modeling clinical care information, ArcherERCASE provides the following entities types: *Admin_Entry, Observation, Evaluation, Instruction, Action and Activity*. All those types represent abstractions of clinical concepts found in a Health domain. One can see in Figure 7 that the entities denoting the concepts of patient clinical care are *Exams, Prescription, History, Evolution and Clinical_Information*. The importance of having modeling constructors that represent such concepts is justified by the following aspects: firstly, it helps in the understanding of how to identify and classify EHR clinical information, and secondly, each instance of a clinical care entity represents a potential archetype that may be reused.

5 Conclusions

This paper proposed a novel conceptual data model, named ArcheER, based on archetypes and dual modeling. The benefits of using dual modeling constructors in the conceptual modeling of health information systems have not been studied so far. The modeling constructors that compose ArcheER and the modeling technique selected for diagrammatic representation were chosen from openEHR specifications.

ArcheER is an extension of the ER data model because this model has been recognized in literature as a simple and efficient approach for the elicitation of data requirements, providing the abstraction required for representing the concepts of archetypes through its graphical notation. As main contributions, we highlight a reduction in the representation of redundant entities and a gain concerning the modeling of entities characterizing knowledge. Also, a CASE modeling tool based on ArcheER was presented and a set of OCL constraints was specified, illustrating how the ArcheER model provides uniqueness to the EHR. The specification of semi-automatic generation of archetypes in ADL from the data requirements modeled by an ArcheER conceptual schema is a possibility for future research.

Acknowledgment

This work was partially supported by Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco (FACEPE), under the grants APQ-0173-1.03/15 and IBPG-0809-1.03/13.

6 References

[1] Elmasri R, Navathe S. B. “Fundamentals of Database Systems”. Addison-Wesley, 6th ed, 2011.

[2] Marco E, Thomas A, Jorg. R, Asuman D, Gokce L. “A Survey and Analysis of Electronic Healthcare Record Standards”; ACM Computing Surveys, pp. 277–315, 2005.

[3] Mu-Hsing K, Tony S, Andre W.K, Elizabeth M.B, Daniel K.G. “Health big data analytics: current perspectives, challenges and potential solutions”; Int. J. Big Data Intelligence, pp.114-126, 2014.

[4] Späth M. B, Grimson J. “Applying the archetype approach to the database of a biobank information management system”; International Journal of Medical Informatics, pp. 1-22, 2010.

[5] Chen R, Klein G. O, Sundvall E, Karlsson D, Åhlfeldt H. “Archetype-based conversion of EHR content models: pilot experience with a regional EHR system”; BMC Medical Informatics and Decision Making, pp. 9-33, 2009.

[6] Garde S, Hovenga E, Buck J, Knaup P. “Expressing clinical data sets with openEHR archetypes: A solid basis for ubiquitous computing”; International Journal of Medical Informatics, pp.334–341, 2007.

[7] Lezcano L, Miguel A. S, Rodríguez S. C. “Integrating reasoning and clinical archetypes using OWL ontologies and SWRL rules”; Journal of Biomedical Informatics, pp. 1-11, 2010.

[8] Oriol X, Teniente E, Tort A. “Fixing Up Non-executable Operations in UML/OCL Conceptual Schemas”; In: LNCS. Vol.8824, pp. 253-496, 2014.

[9] Dinu V, Nadkarni P. “Guidelines for the Effective Use of Entity-Attribute-Value Modeling for Biomedical Databases”; International Journal of Medical Informatics, pp. 769-779, 2007.

[10] Bernstein K, Bruun R. M, Vingtoft S, Andersen S. K, and Nøhr C. “Modelling and implementing electronic health records in Denmark”; International Journal of Medical Informatic, pp. 213-220, 2005.

[11] Jeffrey A. L, Jeffrey L. S, Blackford M. “Method of Electronic Health Record Documentation and quality of primary care”; J Am Med Inform Assoc, pp.1019-1024, 2012.

[12] Georg D, Judith C, and Christoph R. “Towards plug-and-play integration of archetypes into legacy electronic health record systems: the ArchiMed experience”; BMC Medical Informatics and Decision Making, pp.1-12, 2013.

[13] Bernd B, “Advances and Secure Architectural EHR Approaches”; International Journal of Medical informatics, pp.185-190, 2006.

[14] Martínez C. C, Menárguez T. M, Fernández B. J. T, Maldonado J. A. “A model-driven approach for representing clinical archetypes for Semantic Web environments”; Journal of Biomedical Informatics, pp.150–164, 2009.

[15] Buck J, Garde S, Kohl C. D, Knaup G. P. “Towards a comprehensive electronic patient record to support an innovative individual care concept for premature infants using the openEHR approach”; International Journal of Medical Informatics, pp.521-531, 2009.

[16] Arlow J, Neustadt I. “UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design”. Addison-Wesley, 2nd ed, 2005.