

# A CASE Tool for Modeling Healthcare Applications with Archetypes and Analysis Patterns

André Magno Costa de Araújo<sup>1,\*</sup>, Valéria Cesário Times<sup>1</sup>  
and Marcus Urbano da Silva<sup>1</sup>

<sup>1</sup> Center for Informatics, Federal University of Pernambuco,  
Recife, Brazil  
e-mail: {amca,vct,mus}@cin.ufpe.br

Carlos Andrew Costa Bezerra<sup>2</sup>

CESAR – Recife Center for Advanced Studies and Systems<sup>2</sup>  
Recife, Brazil  
e-mail: andrew@r2asistemas.com.br

**Abstract**— Development of Health Information Systems (HIS) based on dual models allows modifications and extensions to be conducted in the layer of archetypes, reducing dependencies on software developers and on system development tools. However, the literature on HIS has paid little attention to modeling tools that build conceptual data schemes based on dual models and archetypes. This paper proposes a metamodel to represent healthcare concepts and their relationships whose instance is seen as a set of analysis patterns because they are useful to more than a single domain and is a dual conceptual schema based on reusable archetypes. The development of a novel Computer-Aided Software Engineering (CASE) modeling tool is discussed, which is called ArcheERCASE, is based on the metamodel proposed, helps Database (DB) designers in the modeling of HIS applications and enables the reuse of archetypes and the reuse of ArcheERCASE conceptual data schemas. Finally, to illustrate the key features and advantages of the proposed model, an ArcheER conceptual schema built for a real legacy system is discussed.

**Keywords**-Archetypes; Database related software; conceptual data modeling; E-health related software.

## I. INTRODUCTION

The software architecture for HIS proposed by the Open Electronic Health Record (openEHR) foundation aims at developing an open and interoperable computational platform for the health domain [1]. This architecture separates the demographic characteristics of patients and information from the Electronic Health Records (EHR) (called *information level*) from the constraints and standards associated with the clinical data of a specific domain (called *knowledge level*). The dual modeling is the separation between information and knowledge of the openEHR architecture for HIS.

Currently, traditional database modeling techniques, in which both information and knowledge are represented together in a single level schema, are used in the development of many HIS applications [2]. However, HIS must handle a large number of concepts that often change or are specialized after a short period of time and consequently, HIS based on such techniques are expensive to maintain and usually have to be quickly replaced. Therefore, dual modeling approaches to provide conceptual schemas of two-level data (i.e., information and knowledge) are essential. At the first level of the dual modeling, data have no semantics, i.e., their meanings are unknown, and only the data types chosen to represent them are known. The second level

consists of domain-driven definitions represented as archetypes and provided by domain specialists at runtime.

Several studies aimed at validating the use of openEHR specifications in the building of EHR of healthcare applications have been described in the literature [3][4]. However, we identified the lack of appropriate computer tools for supporting the dual modeling of conceptual database schemas to provide an understanding about the problem domain through the concepts of archetypes. This is useful to DB designers in the selection of which archetypes of a given repository satisfy the application needs. Also, the importance of building conceptual data schemas for database applications has been acknowledged for several decades because conceptual schemas provide an abstraction of data requirements and help in the validation of user requirements by facilitating the communication between users and DB designers.

In this paper, we propose a metamodel that describes a set of EHR concepts useful for the design of conceptual schemas of HIS applications. This metamodel contains a set of abstract classes that represent clinical care, knowledge data, patient demographic information and administrative data of a health service provider organization. The instances of these classes are seen as a set of analysis patterns because they may be useful to more than a single healthcare domain and compose conceptual data schemas of HIS applications.

Another contribution of this paper is a CASE modeling tool, called ArcheERCASE, for helping DB designers in the modeling of HIS applications. This tool is based on the ArcheER metamodel proposed here and aimed at: (i) providing users with dual modeling constructors to ensure the modeling of unique EHR, (ii) exploiting the advantages of archetypes of the openEHR specifications to facilitate interoperability among HIS, (iii) using concepts of analysis patterns to enable the reuse of archetyped conceptual schemas in different healthcare applications and (iv) providing graphic interface features to guarantee the sharing of archetypes by importing them from the openEHR public repository and exporting them in Extensible Markup Language (XML) format.

This paper is organized as follows. Section II lists the basic concepts used throughout the article. Section III contains a metamodel for the specification of archetyped and reusable conceptual schemas, the main features of our ArcheERCASE modeling tool and examples of application of this tool to illustrate how archetypes and analysis patterns are reused and shared among different healthcare

applications. Section IV describes the main difficulties encountered in modeling HIS with the use of traditional approaches and the advantages of modeling HIS using the ArcheER. Finally, Section V concludes the paper and highlights future work.

## II. THE DUAL MODELING AND ARCHETYPES

In this section, we describe the main concepts that are essential to understand our ArcheERCASE proposal. In Section A, the definition of archetypes is given, while Section B outlines the main issues related to analysis patterns. Finally, Section C describes the related works.

### A. Archetypes

The main feature of a dual model is the representation of data based on archetypes. Archetypes denote a formal model and a reusable domain concept [5]. Thus, if information is represented as archetypes, it can be shared and extended to be used in many different application areas. Archetypes allow HIS to be built based on specific formalisms of health area, promote semantic interoperability of data and adapt to changes and developments in the health field.

The development of computer systems based on dual models allows modifications and extensions (evolution of clinical concepts) to be conducted in the layer of archetypes, reducing dependencies on software developers and on development tools for computer systems. Alterations and extensions are carried out by means of templates. Templates represent user interaction models to group and extend archetypes [6]. The archetypes can be described in Archetype Definition Language (ADL) [7] or XML.

### B. Analysis Patterns

Reuse mechanisms may help less experienced DB designers through the reuse of software components by patterns definitions. Analysis patterns is a pattern category, which is seen as a reuse mechanism in the requirement analysis and conceptual modelling areas [8]. In fact, according to [8], analysis patterns is defined as a group of concepts applied to the modeling of domains of problems, i.e., to a single domain or multiple domains, being useful to the reuse of knowledge specified by another designer.

In this paper, analysis pattern is used for obtaining the reuse of part of a conceptual data schema or of the entire conceptual scheme of data. Also, the concept of analysis patterns is applied to our work to enable the reuse of a specific modeling constructor of a given conceptual schema or the reuse of openEHR archetypes stored in public domain repositories. To the best of our knowledge, the literature on CASE modeling tools and studies about the development of healthcare systems have paid little attention to these issues.

### C. Related Works and Motivation

As indicated in [4], an archetype minimizes the problems of modeling heterogeneity of EHR data and facilitates the standardization of terminologies and constraints for a given health care sector. Several research projects and many applications have been developed from the concept of

archetypes [1][3]. However, some authors exposed the lack of tools and methodologies that would have helped in modeling archetypes in a database [5][9]. 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.

The main goal of such tool is to provide application designers with computer support to assist in the database modeling activities of healthcare applications based on Archetypes and analysis patterns.

## III. THE ARCHEER CASE TOOL

ArcheERCASE is a computational modeling tool that builds conceptual data schemes based on the dual modeling [9]. For the development of this tool, concepts concerning three-layered architectures, analysis patterns and reverse engineering were used. The first concept allowed the separation among the presentation, business and data layers, while analysis patterns was used in the provision of the ArcheERCASE functionality that allows the reuse of an entire conceptual scheme or of the instance of a specific modeling constructor chosen by the DB designer at runtime. Finally, the concept of reverse engineering [10] was applied to interpret the openEHR archetypes specified in XML language, by drawing their main features and converting them into instances of valid modeling constructors of the ArcheERCASE tool.

### A. The ArcheER Metamodel

The ArcheERCASE tool enables the creation of diagrams containing the constructors and stereotypes suggested by the openEHR specifications. From this diagram the user can create its conceptual schema, which supports the class types shown in Fig. 1.

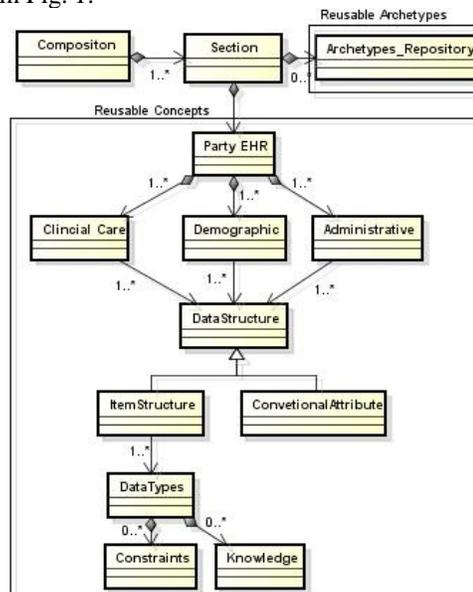


Figure 1. ArcheER Metamodel

Fig. 1 illustrates the ArcheER metamodel specified in the Unified Modeling Language (UML) notation [11]. The *Composition* class represents the metadata of a conceptual schema created by ArcheERCASE and is composed of several sections. A *Section* is the context (e.g., Emergency, Urgency) of a given health field being modeled. ArcheERCASE organizes the modeling constructors of each section in hierarchical structures. The class *EHR Party* expresses the types of information found in EHR and modeled by ArcheERCASE, which are comprised of modeling constructors of the classes *Clinical Care*, *Demographic* and *Administrative*. The *Clinical Care* class denotes all information related to types of assistance and to clinical care given to the patient. The *Demographic* class models information about individuals, groups, organizations or software agents, while the class *Administrative* represents administrative and operational data of a hospital organization.

In addition to constructors for the modeling of EHR, ArcheERCASE has a support for reuse to enable the further use of an archetype specified previously by another designer. The class *Archetypes\_Repository* represents archetypes of public domain repositories that are available and can be incorporated into a conceptual scheme of ArcheERCASE. Note that any redesigned archetype is embedded into a section created a priori by the DB designer.

The class *Data Structure* represents the types of attributes used by ArcheERCASE to specify the information of EHR, and is specialized by the classes *ItemStructure* and *convetionalAttribute*. The first class expresses the attributes (called generic data structures) of archetyped entities (called archetypes), i.e., entities that model clinical care records of patients (e.g., *Clinical Care*, *Demographic* and *Knowledge* entities). The class *ConvventionalAttribute* denotes attributes of the type of entity *Administrative*, i.e., operational information of a service organization in health. For each attribute of an archetyped entity, a data type must be given together with the corresponding terminology and constraints, if any. Thus, the class *Data Type* models the data types specified for each attribute, while the classes *Constraints* and *Knowledge* represent respectively, the constraints and knowledge associated with each attribute of an EHR being modeled. Note that knowledge data is given by a domain specialist and can be a health terminology, an internal vocabulary code or any information of free knowledge.

### B. The System Prototype Architecture

The architecture of a software must have the following components [10]: (i) a layer of user interface, (ii) a management layer for handling objects and business rules, and (iii) a data storage layer. Fig. 2 illustrates the software architecture proposed for ArcheERCASE.

The application layer is responsible for all the functionality of user interaction, providing a set of libraries designed to standardize any graphic environment, and giving the user a better usability.

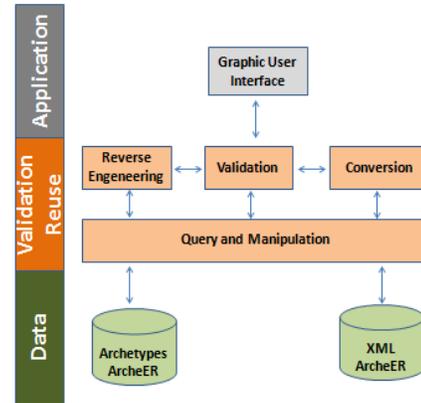


Figure 2. ArcheER Case Architecture

The layer of validation and reuse is responsible for validating all constraints of the ArcheER data model, and is in charge of checking all control structures of the source code and of authenticating all access information to the data layer. Also, the intermediate layer is responsible for providing conversion mechanisms from the openEHR archetypes to conceptual schemas of ArcheERCASE. The data layer provides all the storage structure for the conceptual schema elaborated using our CASE tool.

### C. The Graphic Module of ArcheERCASE

The presentation layer of the software architecture described in Section B corresponds to all components and user interface libraries available in the graphic environment of ArcheERCASE.

As shown in Fig. 3, the graphic environment of this tool provides the following features to the DB designer: i) Commands Menu, ii) Area for building and editing conceptual schemes, iii) Solution Explorer, iv) Properties window and v) Toolbox. All of them are described as follows.

**Commands Menu:** Includes functionality for creating, editing, storing and querying data schemes created with ArcheERCASE.

- **Main Form:** Represents the central area used for displaying, building and editing conceptual schemes, to which modeling components are added.
- **Solution Explorer:** Organizes all components of an ArcheERCASE conceptual scheme by displaying them in a hierarchical structure to facilitate the visualization and handling of all elements of such schema.
- **Properties window:** Allows the designer to describe, edit and view properties of each component.
- **Toolbox:** Provides modeling components for the creation of conceptual schemas, and organizes them according to their respective categories.

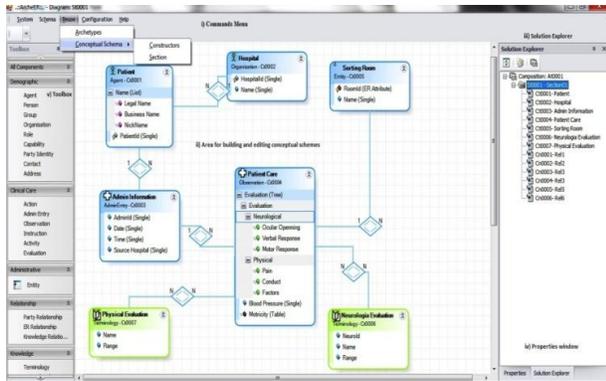


Figure 3. The ArcheERCASE Tool

ArcheERCASE is a graphic drawing software that aims at helping the database designer in his daily data modeling activities by offering him a set of features through a graphic and integrated environment. The main functionality of this tool is detailed as follows.

- Creation of Conceptual Schemas: allows a conceptual schema to be created By default, after the creation of the conceptual scheme, the tool adds the components *Composition* and *Section* to the solution explorer window. If needed, another section may be added to the conceptual schema by the designer.
- Reuse of Conceptual Schemas: allows the reuse of one or all instantiated constructors of a particular *Section* of an ArcheERCASE conceptual scheme.
- Reuse of Archetypes: allows the reuse of archetypes specified in XML that are in public domain repositories. To reuse these archetypes, users, must select an archetype specified in XML. Then, this archetype is inserted into the conceptual scheme previously opened by the designer to reuse its main characteristics, i.e., type and attributes.
- Exportation of Conceptual Schemas: enables the conceptual schema built by ArcheERCASE be exported in two formats. (i) XML format and (ii) an image format (e.g., jpg).
- Generation of Logical Schemas: It generates the logical data schema from the ArcheER conceptual schema.

D. The Reuse mechanisms of ArcheER CASE

The first reuse mechanism of ArcheERCASE converts a openEHR archetype into an instance of a modeling constructor of ArcheERCASE, and this instance is always tied to a Section previously created by the DB designer, while the second mechanism of reuse allows all instances of ArcheERCASE constructors used in the modeling of a particular section be reused in another context of the application (i.e., another section).

To reuse archetypes specified according to the openEHR definitions, ArcheERCASE adopts the concept of reverse software engineering and XML. Fig. 4 gives part of an example of an openEHR archetype written in XML. To validate the XML document and check if it contains an openEHR archetype, ArcheERCASE reads the XML

document and verifies if the tag *xmlns:xsd* of line 2 of Fig. 4 has the signature of openEHR. After validation of the XML document, ArcheERCASE converts the referred archetype into an instance of a valid ArcheER modeling constructor by using the type of the archetype and the data structures that compose it, which are listed in the XML document read previously. To define which ArcheER modeling constructor will be instantiated, ArcheERCASE uses the archetype type definition as shown in line 15 of Fig. 4. Thus, the archetype of the XML document is converted to the notation used by the ArcheERCASE and for the example given in Fig. 4, an instance of the clinical care modeling constructor of type *Admin\_Entry* would be created.

According to the structure of a XML document specified by the openEHR, the definition of data structures that compose an archetype is given by the tag *ontology* as shown in line 16 of Fig. 4. In lines 18 and 23, this figure illustrates that the first two attributes identified by *at0000* and *at0001*, defines the type of the archetype (i.e., *Admin\_Entry*) and the type of generic data structure (i.e., *item\_TREE*), respectively. In lines 28, 33 and 38, the other attributes represent the elements of the *item\_TREE* structures which are imported by ArcheERCASE. Thus, the instance of the *Admin\_Entry* modeling constructor created by ArcheERCASE would have three attributes: Date, Hour and Source.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <archetype xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3 <original_language>
4 <terminology_id>
5 <value>ISO_639-1 </value>
6 </terminology_id>
7 <code_string>pt </code_string>
8 </original_language>
9 <description>
10 <original_author id="name"></original_author>
11 <lifecycle_state>0 </lifecycle_state>
12 <other_details id="MD5-CAM-1.0.1">FF0A2859 </other_details>
13 </description>
14 <archetype_id>
15 <value>openEHR-EHR-ADMIN_ENTRY.example.v1 </value>
16 <ontology>
17 <term_definitions language="pt">
18 <items code="at0000">
19 <items id="text">Example </items>
20 <items id="description">unknown </items>
21 <items id="comment"></items>
22 </items>
23 <items code="at0001">
24 <items id="text">Tree </items>
25 <items id="description">@ internal @ </items>
26 <items id="comment" />
27 </items>
28 <items code="at0002">
29 <items id="text">Data </items>
30 <items id="description">* </items>
31 <items id="comment"></items>
32 </items>
33 <items code="at0003">
34 <items id="text">Hour </items>
35 <items id="description">* </items>
36 <items id="comment"></items>
37 </items>
38 <items code="at0004">
39 <items id="text">Source </items>
40 <items id="description">* </items>
41 <items id="comment"></items>
42 </items>
43 </term_definitions>
44 </ontology>

```

Figure 4. Example of an Archetype in XML

Fig. 5 illustrates the functionality of reuse of archetypes and shows the instance of the *Admin\_Entry* constructor that was derived from importing the archetype written in XML and displayed in Fig. 4.

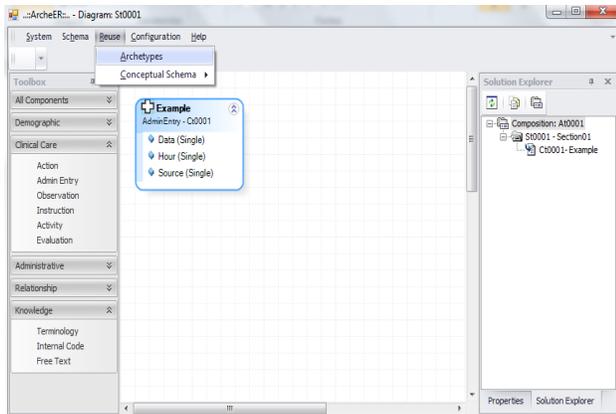


Figure 5. ArcheERCASE Functionality of Reuse of Archetypes

The reuse mechanism of ArcheER conceptual data schemas enables the reuse of one instance or all instances of ArcheERCASE modeling constructors that were used previously in the modeling of a given healthcare application. For example, in the modeling of an *emergency* outpatient care application, the DB designer can reuse a previously created ArcheER conceptual schema to model an ArcheER Section of *urgency*. For this, he must create a new ArcheER Section to represent the current context of *emergency*, and reuse all or some of the instances of the ArcheERCASE constructors previously chosen for the *urgency* section. It is relevant to note the DB designer can reuse the ArcheER conceptual schema entirely that was specified previously by another DB specialist, or create a new version of the previously designed schema, by extending the instances of the ArcheERCASE modeling constructors that are of interest to him.

#### IV. RESULTS

In this section, we describe the main difficulties encountered in modeling HIS with the use of traditional approaches, and later, we comment on the advantages of modeling HIS using the ArcheER. In order to facilitate understanding, we show in Fig. 6 a data schema extracted from a HIS produced by manufacturers of a Health Software in Brazil. The HIS concerns an ambulatory emergency that is performed daily at an Hospital located in Northern Brazil.

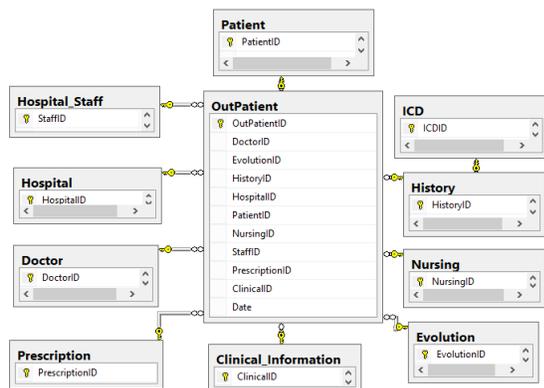


Figure 6. Legacy Data Schema

Observing the data schema, it is possible to see that the initial difficulty happens due to the variety of roles played by the actors in a health domain, such as, workers of a hospital, physicians responsible by patient care, nurses, and other health professionals, that sometimes act as health care providers, and at other times, may be seen as the patient who receives care itself. Besides, the current approaches of 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 Database Management System (DBMS). It is possible to see, in Fig. 6, 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 to each entity, making their information redundant in the EHR.

In the ArcheERCASE, actors are modeled in their more generic way, with new instances being created from the roles played, and therefore, an actor may play several roles in an organization and keeps its record unique. As shown in Fig. 7, 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 some roles, the actor must have training that qualify it for the role, in this case the *Council* entity represents the professional record that the actor needs to have in order to play the role of a physician.

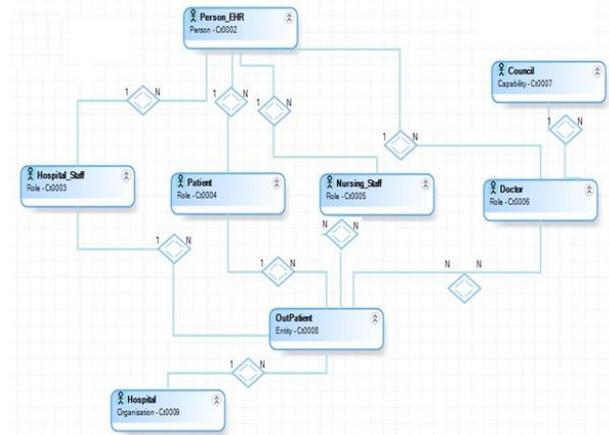


Figure 7. Demographic Conceptual Schema

Besides the roles played in a health domain, an actor can 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* represents the organization responsible by providing services to the patient. Besides the input of demographic information into the EHR modeling, another advantage of the ArcheER model is that, by means of the constraints specified, a demographic entity may only be related with 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 Fig. 8 shows, all relationship with the entity representing patient care (i.e., OutPatient) is being made by means of the roles identified in the described application.

Fig. 8 shows entities that model information of clinical care, administrative and knowledge. 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, and finally, the entity *ICD* represents the terminology used to define the patient diagnosis.

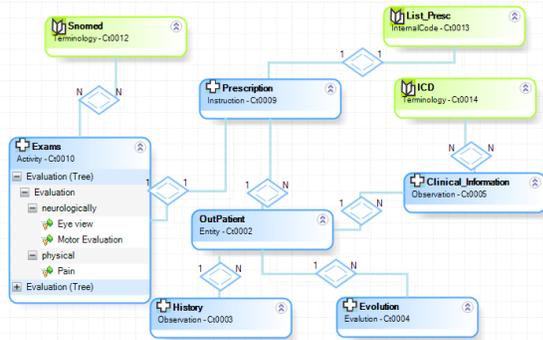


Figure 8. Clinical Conceptual Schema

For the modeling of clinical care information, ArcherER CASE provides the following entities types: *Admin\_Entry*, *Observation*, *Evaluation*, *Instruction*, *Action* and *Evaluation*. All those types represent abstractions of clinical concepts found in a health domain. It is seen in Fig. 8 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.

#### V. FINAL CONSIDERATIONS

This article proposed a novel metamodel to help in the database modeling of HIS through a set of abstract classes whose instances are seen as a set of analysis patterns because they are useful to more than a single domain and help to build dual conceptual schemas based on reusable archetypes. This metamodel enables the generation of dual conceptual database schemas because the demographic characteristics of patients and information from the EHR are modeled separately from the constraints and standards associated with the clinical data of a specific domain.

Aiming at assessing the proposed metamodel, the ArcheERCASE modeling tool was presented, which has the following major contributions: (i) it is based on a reusable collection of analysis patterns denoted by conceptual data schemas generated according to the proposed metamodel; (ii) its data dictionary is stored in XML/ format to allows the

schema exchange and interoperability; (iii) its documentation produced during the project (e.g., conceptual schema and data dictionary) permits further references and visualization, which makes future system maintenance easier; (iv) generates two-level conceptual schemas of data, allowing modifications and extensions to be conducted in the layer of archetypes, reducing dependencies on software developers and on development tools for computer systems. In fact, the impact of using a graphic drawing software in the dual modeling of HIS applications has so far not been studied; and (v) allows the reuse of archetypes specified in XML and stored in public domain repositories. The development of a query language based on the proposed metamodel and the specification of mapping rules to build ArcheER logical schemas object-relational are seen as suggestions of future work.

#### 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.

#### REFERENCES

- [1] C. C. Martínez, T. M. Menárguez, B. J. T. Fernández, and J. A. Maldonado, "A model-driven approach for representing clinical archetypes for Semantic Web environments," *Journal of Biomedical Informatics*, pp.150-164, 2009.
- [2] V. Dinu and Nadkarni P, "Guidelines for the Effective Use of Entity-Attribute-Value Modeling for Biomedical Databases," *International Journal of Medical Informatics*, pp. 769-779, 2007.
- [3] S. Garde, E. Hovenga, J. Buck, and P. Knaup, "Expressing clinical data sets with openEHR archetypes: A solid basis for ubiquitous computing," *International Journal of Medical Informatics*, pp. 334-341, 2007.
- [4] B. Bernd, "Advances and Secure Architectural EHR Approaches," *International Journal of Medical informatics*, pp. 185-190, 2006.
- [5] K. Bernstein , R. M. Bruun, S. Vingtoft, S. K. Andersen, and C. Nøhr, "Modelling and implementing electronic health records in Denmark," *International Journal of Medical Informatic*, pp. 213-220, 2005.
- [6] J. Buck, S. Garde, C. D. Kohl, and G. P. Knaup, "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.
- [7] L. Lezcano, A. S. Miguel, and S. C. Rodríguez, "Integrating reasoning and clinical archetypes using OWL ontologies and SWRL rules," *Journal of Biomedical Informatics*, pp.1-11, 2010.
- [8] M. Fowler, *Analysis Patterns: Reusable Object Models*, Addison-Wesley Professional, 1 ed., 1996.
- [9] M. B. Späth and J. Grimson, "Applying the archetype approach to the database of a biobank information management system," *International Journal of Medical Informatics*, pp. 1-22, 2010.
- [10] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, McGraw-Hill Science/ Engineering/ Math, 7 ed., 2009.
- [11] K. Hamilton and R. Miles, *Learning UML 2.0*, first ed., O'Reilly Media, 2006.