Building Mobile Health Applications Using Archetypes

André Magno Costa de Araújo, Valéria Cesário Times, and Marcus Urbano da Silva
Center for Informatics, Federal University of Pernambuco, Recife, Brazil
e-mail:{amca,vct,mus}@cin.ufpe.br

Carlos Andrew Costa Bezerra
CESAR – Recife Center for Advanced Studies and Systems
Recife, Brazil
e-mail: andrew@r2asistemas.com.br

Abstract — **The use of archetypes in the development of health applications standardizes the data attributes, terminologies, and constraints of the electronic health record, as well as improves the flexibility of health information systems. However, we have noticed in the literature that little attention has been given to researching tools that dynamically build mobile health applications using archetypes, which is what we present in this work - a cloud service for automatic generation of applications from archetypes. The approach hereby proposed specifies a mechanism that generates graphical user interfaces for mobile devices and creates relational data schemes for storing data in the cloud. In addition to that, we present a mobile application named Mobile4EHR that dynamically synchronizes the cloud-generated application with the mobile device, and creates the data schema that allows for local data storage. Finally, aiming to validate the service introduced in this paper, we selected available archetypes in the Open Electronic Health Record Foundation repository to build an application that registers patients' vital signs. The main contributions of our research are i) making the patients' clinical data registration more agile, and ii) reducing the dependence on programmers when creating mobile applications for the health sector.**

Keywords-Graphical user interfaces; Interface generators; Mobile devices and services; Medical informatics; Archetypes.

I. INTRODUCTION

Health Information Systems (HIS) currently play an important role in society. They assist organizations in automating patient care activities, improve the productivity of healthcare professionals, and popularize access to Electronic Health Record (EHR) information. In recent years, the software industry has debated how to develop flexible HIS by harnessing the benefits of cloud computing and the agility provided by mobile devices [1][2]. Cloud computing provides convenient and on-demand access to process and store applications made available in the Internet.

Similar to any other software category, HIS faces problems caused by high maintenance costs, lack of uniformity in EHR modeling, and difficulty in managing the large volume of processed data in the health sector [3].

It is a natural characteristic of software to change over time, and in addition, it must adapt to the new demands of its context, even while it is in execution. HIS deal with a large number of concepts that continually change or are specialized after a short period of time. Consequently, HIS which follow these standards are expensive to maintain. Generally, necessary changes in an application require effort and provoke a high dependence on programming teams. On the other hand, one may notice that HIS are not designed to support dynamic changes, i.e., they are not flexible systems adaptable to the context of the problem domain, and do not allow end users to create new instances of an application or develop new functionalities.

Among the Health standards that promote interoperability, extensibility, and minimize high maintenance costs in the development of the HIS, the dual modeling model proposed by the Open Electronic Health Record (OpenEHR) Foundation stands out [6]. The openEHR architecture separates the generic characteristics that model the EHR structure, also known as the reference model, of the constraints and patterns associated with clinical data, known as the knowledge model.

The concept of dual modeling consists in the separation between the information contained in the EHR from the knowledge associated with the terminologies of the Health area. The first level of dual modeling involves components related to software development (e.g., data schemes, class diagrams, etc.), while the second level is represented by archetypes and templates. An archetype can be defined as a computational expression represented by Health domain constraints, while templates represent user interaction interfaces created at runtime to validate and ensure that data entered conforms to constraints defined in the archetypes [7].

Recent research studies based on openEHR specifications include the construction of the EHR using and specializing archetypes [8], the development of Computer-Aided Software Engineering (CASE) tools for data schema creation [9], and a study on development patterns for Health computing [10]. Moreover, openEHR archetypes have been used to create Graphical user interfaces (GUI) for web applications [11], to store EHR data in heterogeneous databases [12], and to model the EHR in proprietary database systems [13]. However, one may notice that little attention has been given to researching how to dynamically build applications for mobile devices from archetypes.

This work presents a cloud service that dynamically builds Health apps from archetypes. The approach hereby proposed allows Health professionals to construct and distribute applications according to the following pipeline: a representational state transfer (REST) application program interface (API) extracts the EHR specifications from the archetypes (i.e., data attributes, terminologies, and constraints) and dynamically generates relational data schemes and GUI. After that, the application generated is synchronized with the application named Mobile4EHR, and all the functionality generated through the service is ready to use. It is worth mentioning that, at the moment the synchronization between Mobile4EHR and the application created occurs, the cloud-generated data schema is replicated on the mobile device. Therefore, when there is no network or Internet connection, the application will store the data locally. When the connection is reestablished, the EHR data is automatically synchronized with the cloud data scheme.

There are three main advantages in using the service proposed here. Firstly, Health applications are built dynamically following a standard, which makes EHR requirements uniform. Secondly, through mobile devices, Health professionals can expedite the registration of the patient's clinical data. Finally, the service presented here minimizes dependence on programmers in order to develop Health applications.

The rest of this paper is organized as follows: Section II describes the basic concepts used in this work and provides an analysis of the main related works. Section III presents and describes the service proposed in this article, while Section IV demonstrates the creation of a Health application for Mobile4EHR. Finally, Section V presents the final considerations and suggestions for future work.

II. Background and Related Work

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

A. Archetypes

Several research projects and many applications have been developed from the specifications of the openEHR system architecture and the concept of archetypes [8]-[11]. The openEHR software architecture for HIS is aimed at developing an open, interoperable and computational platform for the Health domain [6]. This architecture separates generic information that represents the structures of the 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.

Dual modeling is the separation between information and knowledge of health care system architectures. In this approach, the components responsible for modeling the clinical and demographic data of EHR are specified through generic data structures, which are composed of data types, constraints and terminologies.

In an archetype, the specification of attributes is achieved through data entry builders named generic data structures. Such structures allow the representation of EHR data heterogeneity through the following types: ITEM_SINGLE, ITEM_LIST, ITEM_TREE and ITEM_TABLE.

ITEM_SINGLE models a single data attribute such as a patient's weight, height and age. ITEM_LIST groups a set of attributes in a list. A patient's address containing number, street and zip code for example. ITEM_TREE specifies a hierarchical data structure that is logically represented as a tree. It can be used, for instance, to model a patient's physical or neurological evaluations. Finally, ITEM_TABLE models data elements by using columns for field definition and rows for field value respectively. Each attribute of a data structure is characterized by a type of data and can have a related set of associated domain restrictions and terminologies. The terminologies give semantic meaning to clinical data and can be represented as a set of health terms defined by a professional.

B. Cloud Computing

Cloud computing defines every computational environment, which consists of numerous servers, be they physical or virtual, which have the ability to process and store applications, platforms and services made available in the Internet [14]. As its main feature, it provides convenient and on-demand access to a set of configurable computing resources that can be acquired quickly and released with minimal effort regarding configuration or interaction with the service provider [15]. The term *cloud* is a metaphor for the Internet or communication infrastructure among the architectural components, emphasizing an abstraction that conceals from the user all the complexity of the infrastructure and technologies used to offer such services [14].

Another important feature is that the necessary infrastructure for processing, connectivity, and data storage, is hosted by providers (e.g., *Microsoft Azure e Amazon*) specialized in this type of service. Contrariwise, in a traditional Health domain environment, in order to build or manage a HIS, IT professionals must consider the development, installation, configuration, and software update, aside from other expenses such as software licenses.

The architectural outline of cloud computing consists of three layers: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). The definition of each layer is given below:

- **Software as a Service:** In this layer, software is offered as a service or on-demand. The software runs on a remote server and there is no need to install the application on the client's computer, just access it over the Internet.
- **Plataform as a Service:** A feature provided by the cloud that enables IT professionals to port, within

the cloud, applications built using the programming languages and tools available in the cloud.

- **Infrastructure as a Service:** It consists in the provision of infrastructure for processing, storage, networks, among others. This service, like the others, has its resources shared with several users simultaneously.

The development of a service which generates data schemes and GUI using archetypes is one of the topics encompassed in this study on cloud computing. The service aims at creating relational data schemes and storing EHR in a cloud platform.

C. *Related Works and Motivation*

Based on the main works investigated in the state of the art, this section presents an analysis of the main characteristics of each study, and discusses the main contributions of the work hereby proposed. In order to facilitate the understanding, Table 1 presents a comparative table with four significant criteria that guide the comparison of existing works in the state of the art. The criteria evaluated were: C1) generation of graphical interfaces using archetypes; C2) generation of data schemes; C3) EHR storage in the cloud; and C4) support for mobile applications.

TABLE 1. COMPARATIVE ANALYSIS OF RELATED WORKS

Tool	C1	C2	C3	C4
Template Design	✔	✖	✖	✖
EhRScape Framework	✔	✔	✖	✖
EhRGen Framework	✔	✔	✖	✖
Mobile4EHR	✔	✔	✔	✔

Considering the solutions available in the market, we identified three tools that generate GUIs for the Health sector using archetypes. The Template Design tool, and the EhrScape and EhRGen frameworks support the development of Health applications based on the specifications by openEHR. As shown in Table 1, Template Design, EhrScape, EhRGen, and Mobile4EHR generate their GUIs from archetypes; nevertheless, only EhRScape, EhRGen, and Mobile4EHR generate data schemes and offer the ability to manipulate data in the generated GUI. Lastly, features of EHR storage in the cloud and generation of GUIs for mobile applications is only available on Mobile4EHR. Indeed, the creation of relational data schemes for EHR storage and GUI generation using archetypes is one of the main contributions of the present work. Mobile4EHR is an extension of the GUI generation and customization approach proposed in [33], i.e., it extends GUI generation to mobile applications and proposes to store EHRs in relational data schemes in the cloud.

The main motivation of the proposed work is to develop a cloud service that dynamically builds Health apps from archetypes. For this, we specify a REST API that extracts the EHR specifications from the archetypes (i.e., data attributes, terminologies, and constraints) and dynamically generates relational data schemes and GUI. After that, the application generated is synchronized with the mobile device, and all the functionality generated through the service is ready to use.

III. The Proposal

This section introduces the Health tool generation service from archetypes and is organized as follows: subsection III-A describes the architecture and main features developed, while subsection III-B details the generation of relational data schemes for EHR storage.

A. *Architecture and Overview*

The cloud service proposed in this work consists in a computational environment focused on the dynamic development of applications using archetypes. As mentioned previously, we extend the approach proposed in [11] in the following aspects: Firstly, we modify the GUI generation algorithm to support mobile device usage. Secondly, we have developed a cloud service (i.e., REST API) from the extended algorithm, in order to generate the GUIs. Finally, we developed a mechanism for generating cloud data schemes to persist the EHR data from the generated GUIs.

As shown in Figure 1, by taking advantage of the cloud service, Healthcare professionals can import archetypes and build apps to be used in the Healthcare industry.

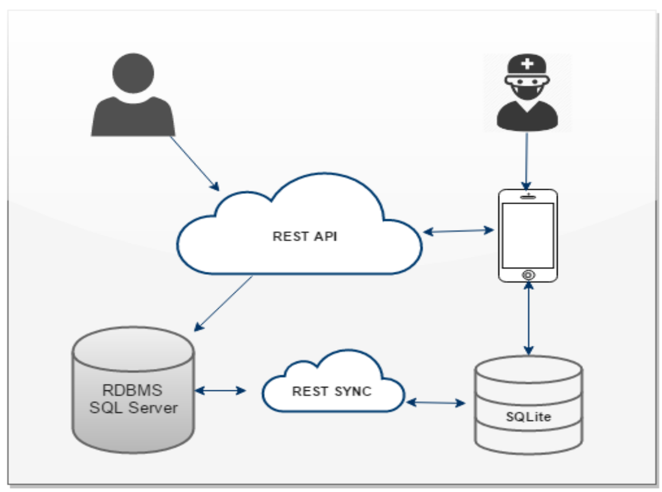

Figure 1. Cloud service architecture for generating apps

To achieve this, a REST API generates relational data schemes and GUI from the extraction of data attributes, terminologies, and constraints from archetypes. As shown in Figure 2, while running Mobile4EHR, the application created is synchronized in the mobile device, and all the features generated in the cloud are available for use. At the time of synchronization, the relational data schema created is replicated in the mobile device. Mobile4EHR stores data locally when there is no network or Internet connection. Once the application identifies a connection, the data is automatically synchronized and the data is uploaded to the cloud.

Figure 2. Interface of Mobile4EHR

In addition to GUI and data schema generation features, the service proposed here counts with the following features: **Demographic Information Management**: This feature allows for managing the actors who will be directly connected to the use of the generated app. In this case, it is possible to manage and link the organization providing the Health services and the professionals (e.g., physicians, nurses and technicians) responsible for the caring for the patient. **Domain Creation:** An organization may offer various types of health services to society. For example, a hospital can perform laboratory tests, diagnostic imaging, emergency care, hospitalization, among others. Having that in mind, Mobile4EHR allows the user to create and configure domains and subdomains that represent the services offered by each organization. This enables to link and access generated GUIs through domains and subdomains.

B. *Relational Data Schema Generation*

The relational data schema created contains 6 tables, 5 integrity constraints and a set of fields, which store the extracted elements from archetypes. The *Archetype* table stores the informed XML configuration file metadata, e.g., the type of archetype, its author, the file version, among other information. The *Archetype_Details* table stores the type of the data structure and constraints found in the archetype, while the *Terminology* table stores reference data attributes alongside their respective terminologies.

An archetype data specification may be done via a single attribute, a vertical list of attributes, a hierarchical data structure or a table with rows and columns. Aiming to store the data attributes into an archetype while respecting all possible layouts, we use the *ITEM_TREE* and *ITEM_TABLE* tables. Since the organization of a hierarchical structure already includes the definition of one or multiple attributes, we have mapped the attributes of the SINGLE, LIST and TREE types to the ITEM_TREE table. As the name suggests, the ITEM_TABLE table maps the attributes of the archetypes that are arranged in rows and columns. Once all data attributes are mapped to their respective tables, the user can choose which data attributes will generate the GUI, and therefore store the data manipulated by end-users. In this case, the selected attributes are added at runtime as columns in the Data_Item table.

Each element of an archetype is referenced by an identifier. All identifiers begin with the letters *at* followed by a sequential value. For example, the *at000* identifier stores the name given to an archetype. We have applied the naming standard defined by this identifier to name the fields in our data schema. Whenever a new field is inserted in the *Data_Item* table, the service verifies whether it already exists, avoiding repetitions.

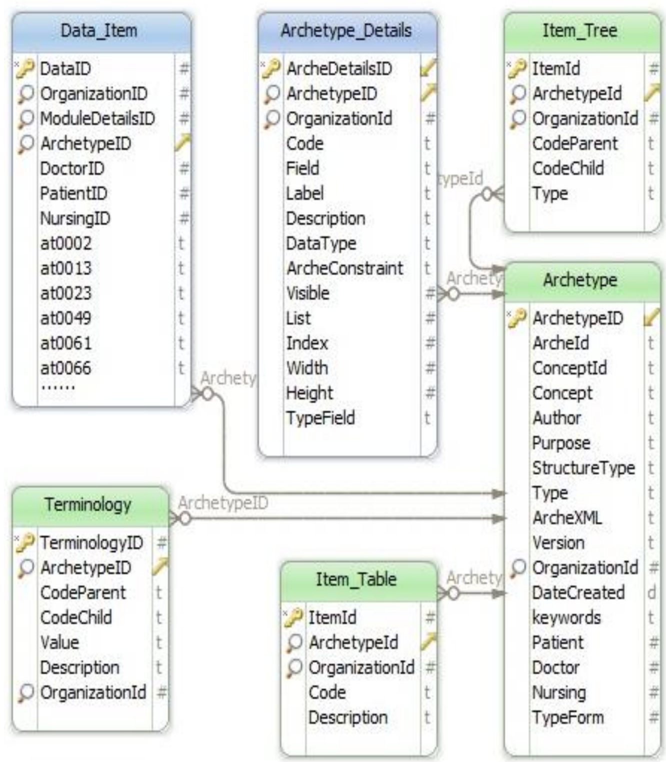

Figure 3. Relational data schema created from archetypes

Figure 3 shows a data schema instance built from Blood Pressure and Apgar archetypes, both available at the openEHR repository.

IV. RESULTS

In this section, we demonstrate the generation of an app for the Healthcare industry using the service described in this article. Our goal is to exemplify the activities carried out by a Health professional, such as collecting and registering patients' vital signs. We chose three archetypes related to such activities, all available in the openEHR repository: Blood Pressure, Body Temperature and Respiration.

First of all, a Health unit named Hospital was registered. Subsequently, we registered a domain called Vital Signs and three subdomains named Blood Pressure, Body Temperature and Respirations, which will later be linked to the GUI generated from the archetypes. Afterwards, the archetypes were imported, linked to their respective subdomains and the application was released and available to be used. At that moment, opening Mobile4EHR triggers the synchronization and installs the cloud-generated service. Figure 4 shows the domain (i.e., Vital Signs) and subdomains (i.e., Blood Pressure, Body Temperature and Respirations) created to access the GUIs created in this demonstration, while Figure 5 depicts the GUI created from the Body Temperature archetype.

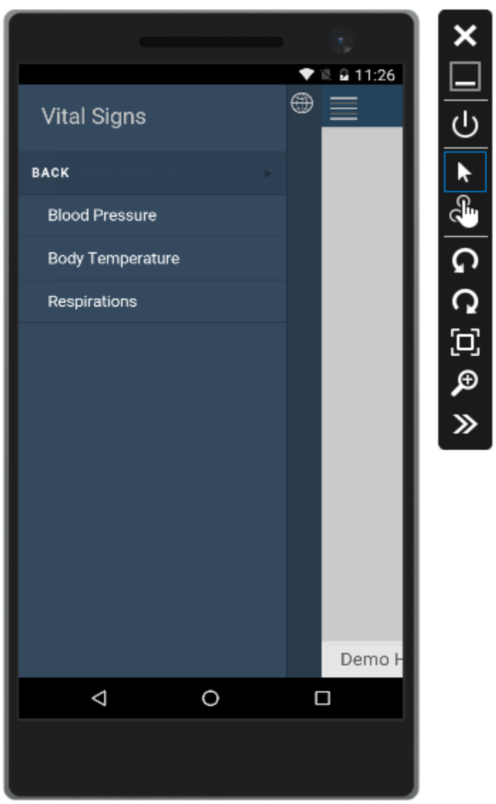

Figure 4. Menu of the app created

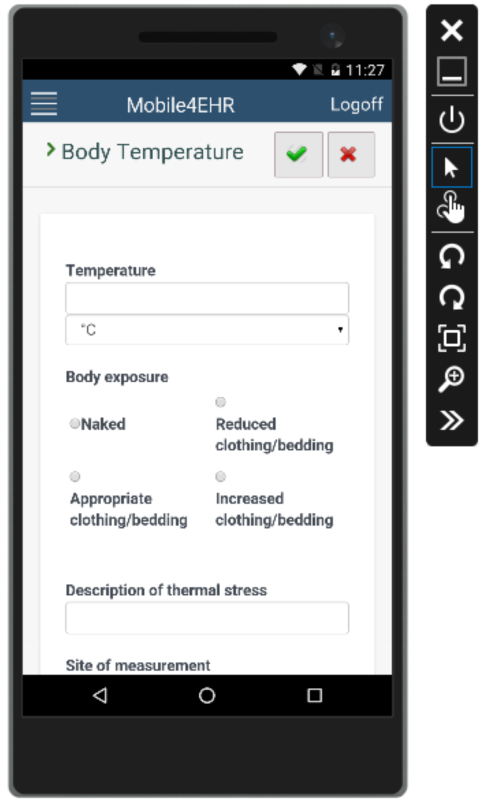

Figure 5. GUI created from the archetype Body Temperature

Each generated GUI has data persistence features. (i.e., insert, update, delete, and select). The data manipulated from the GUI is stored in the local relational data schema or in the cloud.


```
[
  {
    "Temperature": "33°C",
    "BodyExposure": "Reduced clothing/bedding",
    "DescriptionThermalStress": "",
    "Measurement": "Axilla"
  },
  {
    "Temperature": "39°C",
    "BodyExposure": "Appropriate clothing/bedding",
    "DescriptionThermalStress": "",
    "Measurement": "Urinary bladder"
  }
]
```


Figure 6. Data synchronization

Figure 6 portrays the records of the local data schema in JSON format, extracted through the REST API responsible for synchronizing data with the cloud database. In this task, only records that are in the local database and not in the cloud are synchronized. This validation aims to minimize application performance problems.

 ISBN: 978-1-61208-538-8

V. FINAL CONSIDERATIONS

This article presented a cloud service that enables users to build mobile Health applications using archetypes. As the main contributions, we highlight: i) we created a GUI generation service for mobile devices; ii) we have specified a mechanism for generating cloud relational data schemes for EHR storage; iii) we developed an app called Mobile4EHR that synchronizes the application created in the cloud, and generates the data schema locally on the mobile device. Furthermore, we have demonstrated the creation of a Health app from openEHR archetypes to register patients' vital signs.

Four main characteristics stand out in our work. Firstly, Health applications are built dynamically from a standard that makes EHR requirements uniform. Secondly, through mobile devices, Health professionals can speed up the recording of a patient's clinical data. Thirdly, the EHR is stored on a platform in the cloud. Finally, the approach proposed here minimizes the dependence on programmers in order to develop Health applications.

The development of a mechanism for generating NoSQL cloud data schemes and the evaluation of usability tests with health professionals are the next aims of our future works. In this paper, we limited the scope of research to data schemes and GUIs generation. A forthcoming work will address privacy and security issues by presenting an algorithm to encrypt EHR data on a cloud service or local storage.

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 P. Nadkarni, "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.343-353, 2010.

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

[9] A. M. C. Araujo, V. C. Times, and M. U. Silva, "A CASE Tool for Modeling Healthcare Applications with Archetypes and Analysis Patterns," The Eleventh International Conference on Software Engineering Advances, pp. 206-211, 2016.

[10] 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. 205-226, 2010.

[11] A. M. C. Araujo, V. C. Times, and M. U. Silva, "Template4EHR: Building Dynamically GUIs for the Electronic Health Records Using Archetypes," 16th IEEE International Conference on Computer and Information Technology, pp. 26-33, 2016.

[12] A.M.C Araujo, V.C Times, and M.U Silva, "PolyEHR: A Framework for Polyglot Persistence of the Electronic Health Record", Int'l Conf. Internet Computing and Internet of Things, pp. 71-77, 2016.

[13] L. Wang, L. Min, X. Lu, and H. Duan, "Archetype relational mapping - a practical openEHR persistence solution," BMC Medical Informatics and Decision Making, pp. 1-18, 2015.

[14] R. S. Seelam, P. Dettori, P. Westerink, and B. B. Yang, "Polyglot Application Auto Scaling Service for Platform As A Service Cloud," IEEE International Conference on Cloud Engineering, pp. 84-91, 2015.

[15] M. Ellison, R. Calinescu, and R. Paige, "Re-engineering the Database Layer of Legacy Applications for Scalable Cloud Deployment," IEEE/ACM 7th International Conference on Utility and Cloud Computing, pp. 976-979, 2014.