

A Model-Driven Based Methodology for the Generation of Context-Aware Medical Interfaces from Open EHR Archetypes

Afef Samet Ellouzea^{1*}, Sandra Hammami Tilia¹ and Rafik Bouaziz²

¹Higher Institute of Technological Studies, ISET, Sfax, Tunisia

²Faculty of Economic Sciences and Management, Sfax University, Tunisia

Abstract

Purpose: Different medical computer science standards, such as ISO/EN 13606, openEHR and HL7 CDA, use the dual model approach to support interoperability between medical information systems. However, use of these standards encounters some obstacles: (i) lack methodological tools to facilitate design of Electronic Health Record (EHR) systems in accordance to two conceptual levels, (ii) Lack of methodologies to perform semantic activities on EHR content in the semantic web environment, and (iii) Absence of models to adapt medical knowledge, modelled by means of archetypes, in different contexts of use. This paper presents a model-driven based methodology for the generation of context-aware medical interfaces. This methodology, entitled OpenEHR modelling Methodology: OpenEHR-MM includes concepts, models, rules, processes and tools aimed to assist EHR designer in the generation process of context-aware medical interfaces.

Methods: OpenEHR-MM is based on the use of the model driven architecture technological space, and it includes the following four steps:

1. Analysis and specification of requirements: This first step aims to guide EHR interface designer in the definition of users' needs.
2. Conceptualization: This step aims to define tools, concepts and approaches that assist EHR designer in the two conceptual levels modelling process. At this step, we define an approach to provide the context information modelling.
3. Ontologization: This step aims to assist the designer to integrate the semantic dimension in the EHR systems modeled in accordance to the dual model approach.
4. Implementation: Finally, OpenEHR-MM provides an adaptive approach to enable medical interface configuration with respect to contexts settings.

Results: A model driven-based methodology for context-aware medical interfaces generation process. The experimentation of this methodology is performed on the EHR related to patients affected by cerebral palsy.

Keywords: Electronic medical record; Context-aware interface; Model driven architecture; Semantic interoperability; Open EHR standard

Introduction

The practice of medicine requires collaboration between several care providers. Each actor intervenes according to its role and its competence for the process of patient data. However, each care provider cannot be restricted only to data which he has introduced itself. Indeed, he needs also an efficient sharing of data introduced by other actors. Hence, the role of the Medical Information Systems (MIS) is to ensure this data sharing and to provide relevant information which can be adapted to user context. Use context adaptation consists on the capacity of the MIS to be able to adjust its behaviours, so that it can be used in an optimal way in any environment described by a specific contextual situation. However, clinical information is generally allocated to several independent Electronic Health Records (EHR) systems, which are usually syntactically and/or semantically incompatible. This fact precludes healthcare professionals from accessing patient's clinical information in a comprehensible way and hinders adaptation of this information to the user contexts. Therefore, it is appropriate to provide semantic interoperability of EHR systems in order to ensure context-aware MIS.

In this setting, the ISO/TR 20514 standard [1] stipulates that four conditions must be met for the semantic interoperability of EHR:

- (i) Agreement on a standardized reference model,
- (ii) Standardized interface models services to ensure interoperability between different healthcare services,
- (iii) Standardized set of concepts for domain-specific models, and
- (iv) Standardized terminology associated with controlled vocabularies.

However, these conditions are difficult to satisfy by traditional information models. Indeed, the classic modelling of EHR is based on the "one-level" architecture, that is to say a single model that gathers information and clinical knowledge. In this type of architecture, domain concepts are hard-coded into the database models. If the classic modelling development of MIS is relatively easy, it turned out that the

*Corresponding author: Afef Samet Ellouze, ISET, Higher Institute of Technological Studies, Road of Mahdia, Km 2.5 Elbustan, W.B. 88A, 3099 Sfax, Tunisia, Tel: +216 74 620 268; Fax: +216 74 214 017; GSM: + 216 27 410 184; E-mail: samet_afef@yahoo.fr

Received July 22, 2017; Accepted July 28, 2017; Published July 31, 2017

Citation: Ellouzea AS, Tilia SH, Bouaziz R (2017) A Model-Driven Based Methodology for the Generation of Context-Aware Medical Interfaces from Open EHR Archetypes. J Health Med Informat 8: 279. doi: [10.4172/2157-7420.1000279](https://doi.org/10.4172/2157-7420.1000279)

Copyright: © 2017 Ellouzea AS, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

context of the exchanged information is difficult to find. This is because the medical language is ambiguous. Indeed, it presents problems of synonymy, i.e., several terms can possibly be used to define the same concept, and problems of polysemy, e.g., the same term can denote several concepts. To ensure the sustainability and efficiency of EHR systems, several research studies have focused on providing models for the storage and structuring of information and medical knowledge. Proposed solutions are based [2,3] on the principle that it is appropriate to have a meta-modelling of EHR systems [4,5]. Thus, the advanced EHR representation and communication standards proposed using an architecture based on the dual model approach. This approach, which is an alternative to the “one-level” modelling one, consisting of separating the semantics associated with information and knowledge in two levels of models [6]. The first one corresponds to the most familiar model for most developers; it is the level of object models and database schemas, called the reference model. This model is used for the construction of the information system. The reference model specifies global and permanent characteristics of EHR components. It also determines how these components can be aggregated as well as the information needed to satisfy legal, ethical, and other requirements. The second level corresponds to the knowledge level, requiring its own formalism (s) and structure (s), and this is where the domain concepts are expressed. It is called the archetype model.

Archetypes are reusable and can be selectively aggregated or refined to meet specific needs, through the use of templates [7]. According to templates are a result of generations in HTML format of archetypes, allowing their customization according to a specific context of use. Templates combine several archetypes, according to a local context, in larger structures, such as forms, documents and reports. These templates can impose other local constraints on archetypes (e.g., deletion or addition of optional sections and definition of default values).

Several current works has focused on the use of archetypes for defining the structure and semantics of EHR. Also, different medical computing standards, such as ISO 21731 “Health informatics - HL7 version 3 Reference Information Model” [8], ISO/EN 13606 “Health Informatics-EHR communication” [9] and openEHR [10], use the dual modelling approach to support interoperability between MIS. However, the archetype model for this modelling approach allows only the definition of the clinical data structures and does not contain information to specify how these structures must be adapted to the user context.

Besides, in the literature several studies have suggested various context modelling approaches based on different models. Chen and Kotz [11] and Strange and Linnhoff-Popien [12] identified six most popular techniques for modelling context: key-value models, tag models, graphic models, logical models, object-oriented models and ontological models [13], which are the followings:

- **Key-value model:** It is considered the simplest approach for the modelling and representation of contextual information. This approach uses key-value pairs to represent information context [13]. The key-value modelling approach is frequently used in distributed service frameworks [14]. Such a simple model is powerful enough to allow pattern-matching queries. Despite the ease of implementation of this model, it can lead to conflicts of interpretation of the context [15]. In particular, key-value pairs lack capabilities for enabling efficient context retrieval algorithms [12].
- **Markup scheme models:** They are based on a fixed hierarchical structure consisting of tags, with attributes, associated with content [7]. In particular, the content of the mark-up tags is usually recursively defined by other markup tags. Typical representatives of this kind of context modelling approach are profiles. They usually base upon a serialization of a derivative of SGML [15]. Most of these models are derived from the Composite Capabilities/Preference Profiles 14 (CC/PP) model that been proposed by the W3C to address problems of access to pages or services web by increasingly scalable devices (smart phones, tablets, connected TVs, etc.). The advantage of these tag models is the use of reusable XML schemas for defining contextual properties. However, these models are limited in term of expressivity. In particular: the expression of constraints on the values of the properties and the expression of the relations and links between these properties [13].
- **Logic-based model:** Logic defines the conditions on which a concluding expression or fact may be derived from a set of other expressions or facts. In a logic based context model, the context is consequently defined as facts, expressions and rules. This model uses the facts, expressions and rules to model the context. The use of rules allowing the expression of different policies, constraints and preferences offers a rich expressiveness compared to other models. However, the use of these models is reduced because of the lack of standards [13].
- **Object-oriented models:** These models use the object-oriented approach contextual information because they are based on a location object and a navigation point object. The contextual information is embedded as the states of the object, and the object provides methods to access and modify the states. Object-oriented context modelling approaches benefit from its advantages encapsulation and reusability [13]. However, these models suffer from the lack of reasoning abilities and the lack of standards [13].
- **Graphical models:** These models are based on formal graphical languages. Unified Modelling Language (UML) and Object Role Modelling (ORM) are examples [13]. In ORM, the basic modelling concept is the fact, and the modelling of a domain using ORM involves identifying appropriate fact types and the roles that entity types plays [12]. The wide acceptance and use of UML in the software engineering domain is one of the main motivations for using graphical approaches for context modelling and representation. One of the most prominent advantages of graphical approaches is their strength in structuring based on human-friendly visual constructs [4]. One of the principal advantages of these approaches is their force in the structuring of contextual information. In this perspective, “OCL” language (Object Constraint Language) is used to provide support of reasoning for the modeled contextual information due to class diagrams’ UML. However, the expressiveness and reasoning functions supported by conjunction of OCL to UML remain limited compared to the intake of ontology.
- **Ontology based model:** Ontology provides a strong source of semantic reasoning and conflict resolution in context interpretation [16]. Ontology is a promising instrument to specify concepts and interrelations. It is particularly suitable to project parts of the information describing and being used in our daily life onto a data structure utilizable by computers [12]. Ontology-based model combine logic and object-oriented

modeling techniques. Accordingly, context is characterized by classes (concepts), individuals (facts) and properties (roles, relationships). More complex concepts are usually defined by derivation from simpler concepts. Similar to the case of the logic-based models, the high level of formality and support for reasoning are the main advantages of the ontology-based context modelling technique. Furthermore, the capabilities of knowledge sharing and knowledge reusing additionally underpin the usage of the approach [17]. Note that because of the many advantages of the ontology-based approach. Several researchers believe that this is the most appropriate way to represent and manage the context. In Benslimane and Arara [12,18,19], consider that the context is closely related to the notions of level of abstraction, granularity and interest of user communities and perception of ontology developers. They consider that the same domain can be modeled by several single-contextual ontology, where each one is described in a particular context.

Noting that, the proposed context modelling approaches focused only on the “one-level” architecture based systems. The purpose of this study is to define a methodology for the design and development of EHR systems modeled according to the openEHR standard, as well as the generation of context-aware medical interfaces from these systems. Our adoption of this standard is argued by the following reasons:

- (i) openEHR has a large users and developers communities and is used in several countries, such as Australia and Holland;
- (ii) Several open source tools for editing and validating archetypes and templates are available for this standard;
- (iii) An openEHR clinical knowledge manager, called the Clinical Knowledge Manager (CKM) [20], is also available online for this standard. This methodology is entitled OpenEHR-MM: OpenEHR Modelling Methodology. As part of this study, we have opted for (i) first, modelling contextual information through UML, (ii) Then transform class diagram to ontological schema [21] in order to benefit from added value by using ontology.

We applied OpenEHR-MM to the case of EHR related to patients affected by Cerebral Palsy (CP), in response to the needs felt by physicians treating this type of chronic disease. Indeed, this disease requires a multidisciplinary follow-up and an efficient sharing of data between the different specialists involved in the therapeutic treatment course of patients.

Methods

Global approach

OpenEHR-MM includes concepts, models, rules, processes and tools that we provided, combined and/or adapted to assist designers and developers in the generation process of context-aware medical interfaces from EHR systems modeled in accordance to the openEHR standard. Note that, we exploited the Model Driven Architecture (MDA) technological space to take advantage of the architecture’s contributions, in terms of models levels abstraction, independence of descriptions and transformation. Indeed, we implement in our methodology a model construction according to the three layers of MDA, which are the Computation Independent Model (CIM), the Platform Independent Model (PIM) and the Platform Specific Model (PSM). Then, we ensure the direct models transformations between

these different layers. Regarding the expression of models, we opt to use UML as a powerful modelling language, well suited to the MDA technological space. Our choice is argued by the following reasons:

- UML is a graphical notation produced as a result of many years of experience in software analysis and design by a variety of companies in a wide range of industries and domains;
- UML is widely adopted in the software industry and taught in many university courses;
- UML is an open standard held by the Object Management Group (OMG);
- UML is a polyvalent language because it allows the modelling of different views of a software system, using several diagrams such as the class diagram (to represent the structural description of the system) and the component diagram (to represent the various modules of the system and to highlight the dependencies between different components).

In particular, OpenEHR-MM assists in the development and implementation of a context-aware medical interfaces generation process. This article deals more with this aspect.

Concepts, models and building rules of OpenEHR-MM

The development of OpenEHR-MM requires a combination of several concepts, models and rules that we adopt from: (i) The UML object oriented modelling language, and (ii) The dual modelling paradigm related to the openEHR standard. Regarding UML, OpenEHR-MM proposes the use of a set of UML diagrams in order to represent the different viewpoints of the EHR system, and to model its specific concepts. UML diagrams are as follows:

- At the level of the functional model, we exploit the UML use case diagram, which allows needs expression in terms of services provided by the system. Via this diagram, designer can define two main components: actors and use cases. An actor (e.g., a doctor, a patient) is an external entity of the system that can initiate one of the uses cases. A use case is a feature provided by the system (e.g., creation of an EHR).
- At the level of the structural model, we used the UML class diagram to model the stable (non-volatile) concepts of an EHR system. This diagram allows the designer to represent the structural aspect of a system.

Concerning the modelling environment specific to the openEHR standard, openEHR-MM adopt the use of the two models proposed by this standard: the reference model and the archetype model, respectively for modelling information and clinical knowledge. At this level, we decide to represent the OpenEHR-MM class diagram by the use of two main packages. The first package is used to model the different aspects underlying the information model (e.g., the stable entities of an EHR), and the second package is used to specify the components of the archetypes required for EHR modelling (e.g., knowledge). To this end, we have taken back the use of the “archetype” package, relative to the standard reference model of the standard openEHR, in order to specify the archetypes constraining the information of any EHR [22]. The “archetype” package groups the classes “LOCATABLE”, “PATHABLE” and “ARCHETYPED” used to reference a well-defined archetype (e.g., “openEHR-EHR-OBSERVATION.blood_pressure.v1” is a reference of the archetype describing blood pressure).

However, archetypes only allow the definition of clinical data structures and do not include means to indicate how these structures should be adapted to the context of the user. Thereby, we see the need to extend the package “archetyped”, of the reference model openEHR. In order to support knowledge related to the presentation and the contents of the data returned, according to the context of the user.

For this purpose, we introduce a new concept that we entitle “Context-aware reference model”. We model the package “archetyped”, which we rename “CONTEXT-AWARE ARCHETYPED”, corresponding to the context-aware reference model, through the following classes:

- “ARCHETYPED”: It corresponds to the class “ARCHETYPED” contained in the standard reference model of the standard openEHR [22]. However, we decide to add the attribute “Path”. In addition to the attribute “Archetype Reference” initially present at this class. The attribute “Path” describes the path of an archetype in archetypes’ library constructed in order to regroup the concepts required for modelling an EHR.
- “VIEW”: This is an association class that links the class “ARCHETYPED” to the class “CONTEXT”. We decide to integrate: (i) The class “VIEW” in the package “archetyped” and (ii) The class “CONTEXT” in the package “entry”. The class “CONTEXT” describes a user’s context settings. The class “VIEW” describes the different views of an archetype, which can vary according to the context parameters we have structured in the class “CONTEXT”. We define the class “VIEW” by the following attributes: “ViewId”, “ViewName” and “ViewDescription”. Regarding the class “CONTEXT”, we define it by the attributes “ContextId”, “Specialty”, “AccessRight”, “Skill” and “Preferences”. We also add the following temporal attributes to this class: “ContextStartTime” and “ContextEndTime”. These attributes are used to specify the application interval of the context.
- “UI_Template”: This class describes the user interface characteristics. A user interface is formed by the views of the different archetypes, according to a well-defined context. These views are modeled by the class “VIEW”. Each view of an archetype is associated with a user interface component, called “widget”. Widgets are presented in a specific area, depending on the adaptation process. We describe the class “UI_Template” with the attributes “UI_TemplateId”, “UI_TemplateName” and “UI_TemplateDescription”.
- “WIDGET”: It represents the elements making the user interface, such as textarea, list and checkbox. We describe this class using the attributes “WidgetId”, “WidgetName” and “WidgetType”.
- “ZONE”: This class is used to describe the presentation parameters associated with each widget. It is described using the attributes “ZoneId” and “ZoneOrientation”.

Figure 1 illustrates the package “CONTEXT-AWARE ARCHETYPED” corresponding to the context-aware reference model we propose. Note that the incomplete elements in this figure corresponding to those present in the standard reference model openEHR [22].

Furthermore, OpenEHR-MM includes some rules that lead to the construction of archetypes, including the following:

- **Rule 1:** All information elements, e.g., items to be documented, collected from existing paper-based or electronic-based systems, must be captured and analysed according to their structure and context.
- **Rule 2:** All items obtained in accordance with “Rule 1” must be structured and combined with distinct concepts, which do not overlap.
- **Rule 3:** All concepts obtained in accordance with “Rule 2” must be transformed into archetypes.
- **Rule 4:** Each archetype found must represent a comprehensive concept of clinical knowledge, thus covering the maximum possible data combination.
- **Rule 5:** The archetypes obtained in accordance with “Rule 3” must constrain the information model of an EHR.

General approach of OpenEHR-MM

We synthesize the different steps of OpenEHR-MM as following:

(A) Analysis and specification of requirements: At this step, the EHR designer needs to define user needs through interviews with experts. We use the direct interview method, which consists of questioning the care actors involved in the therapeutic pathway of the considered disease. These interviews allow the designer to determine all items to be documented from existing paper-based or electronic health systems and to understand the meanings and contexts of use of these items. The context groups together a set of parameters encompassing user characteristics, preferences and interests. The census of context parameters helps in the generation of adaptive medical interfaces. This step is modeled by using a UML use case diagram. This representation aims to formalize these needs. Subsequently, it is clear that the requirement specification step represents the CIM layer of the MDA technological space.

(B) Conceptualization: At this phase, the EHR designer must model clinical knowledge by transforming them into archetypes. He must also design an EHR information model in accordance with the openEHR reference model [21,23,24]. This step, relating to the PIM layer of the MDA technological space, includes the following actions:

- Structure the items into clinical concepts. This is a harmonization process that brings items into concepts. The concepts identified and the corresponding items must be structured hierarchically; this structure can be visualized, for example, in the form of a conceptual mapping. It is necessary to identify the point of view of the therapists for each concept in order to determine subsequently the behaviour of the associated archetype at the level of each specialty.
- Transform the identified concepts into archetypes existing in open archetypal repositories to form a library of archetypes used to constrain the EMR information model. At this stage, we decide to reuse as much as possible the existing archetypes in the open depot managed by the “Clinical Knowledge Manager” (CKM) tool and to develop new archetypes only when needed. This avoids the development of overlapping archetypes that hinder the interoperability of EHR. This action is based on the following two principles: (i) A single archetype should represent a single clinical concept and (ii) The archetypes should be designed in a generic and reusable manner to promote their use in different contexts and, thus, interoperability of EHR. The openEHR CKM tool is used for

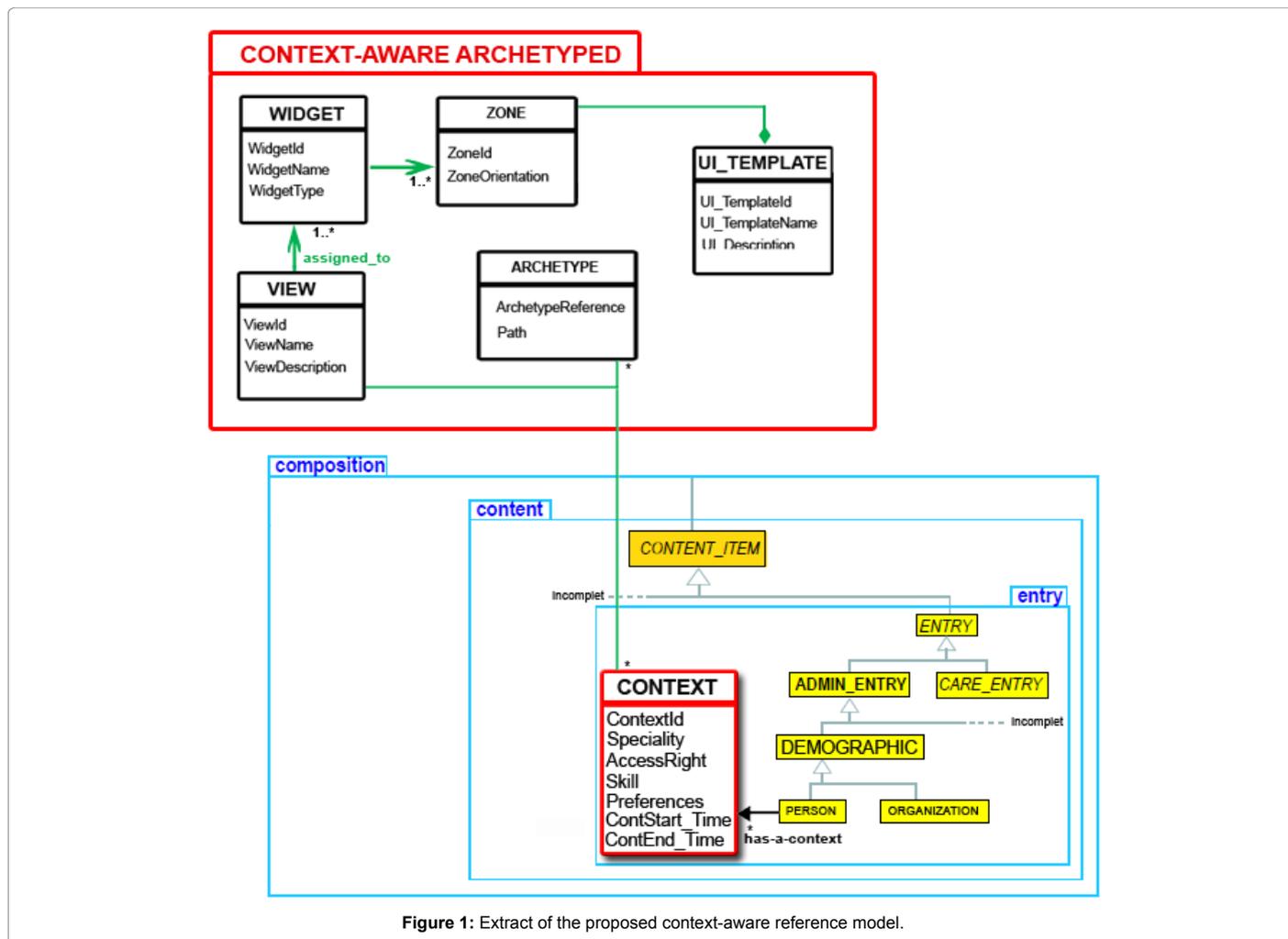


Figure 1: Extract of the proposed context-aware reference model.

the discovery of appropriate archetypes. This tool has several search features (e.g., search by name or by clinical domain), used to find the adequate archetypes.

- Modelling new archetypes for concepts that do not correspond in the open CKM repository. It is also possible that some archetypes should be extended or created, using the editor “Ocean Archetype Editor” [25]. The extension of the archetypes consists in adding the missing items and possibly linking these items to terminological sources, such as SNOMED-CT [26], thus allowing a consistent use of these sources and a suppression of the ambiguity of the clinical terms.
- Modelling the information according to the openEHR reference model. The designed information model corresponds to a UML class diagram [22]. The purpose of this representation is to describe the set of classes used for the modelling of the EHR system. We propose to use the Eclipse open source development platform, which is considered to be the incubator for development projects. It is currently the most widely used platform in the model-driven engineering community for the construction of UML diagrams.

(C) Ontologization: At this step, the EHR designer must integrate the semantic dimension into clinical archetypes to provide mechanisms

for the execution of semantic activities (e.g., search, selection, classification) in EHR systems. This step is carried out accordingly to the methodology described in our previous research work [22]. Indeed, we have proposed a methodology which aims to assist designers and to provide them with tools to successfully integrate the semantic dimension into EHR systems. The process of this methodology is based on three main steps, which are: (i) The expression of archetypes according to OWL-DL, hence, ensuring their semantic definition; this step aims to build an archetype library expressed in accordance with the OWL-DL language. This archetype library must correspond to instances of the openEHR standard knowledge model, which we proposed to be enriched using the semantic dimension; (ii) The creation of an ontological source which can be used to annotate EMR archetypes and to enrich the ontological source by supplying semantic relations in order to satisfy the ontology building process; as well as (iii) The annotation of clinical archetypes which enhances data extraction and possibilities of performing knowledge-intensive tasks on archetyped EHR extracts. Semantic search and comparisons activities will be based on annotated archetypes. Note that the ontologization step requires the cooperation of the MDA space and the semantic web space.

(D) Implementation: At this phase, the EHR designer must generate context-aware medical interfaces. These interfaces correspond to instances of the template of openEHR. Indeed, the templates make it

possible to customize the archetypes according to a context of specific use. This step is based on the use of the open source tool “Template Designer”. It corresponds to the PSM layer of the MDA technological space. We detail in the following sub-section the constituents and tools we have adopt for the generation of context-aware medical interfaces.

Figure 2 illustrates the methodological approach of OpenEHR-MM by presenting the main steps performed within each space as well as the interactions between these spaces.

Proposed components and adaptation tools

The context parameters cover several criteria that can influence the configuration of medical interfaces. We consider the following parameters:

- The role: It corresponds to the user’s profile, e.g., a doctor, a nurse.
- The specialty: It describes the clinical specialty of a health actor, e.g., neurology and occupational therapy.

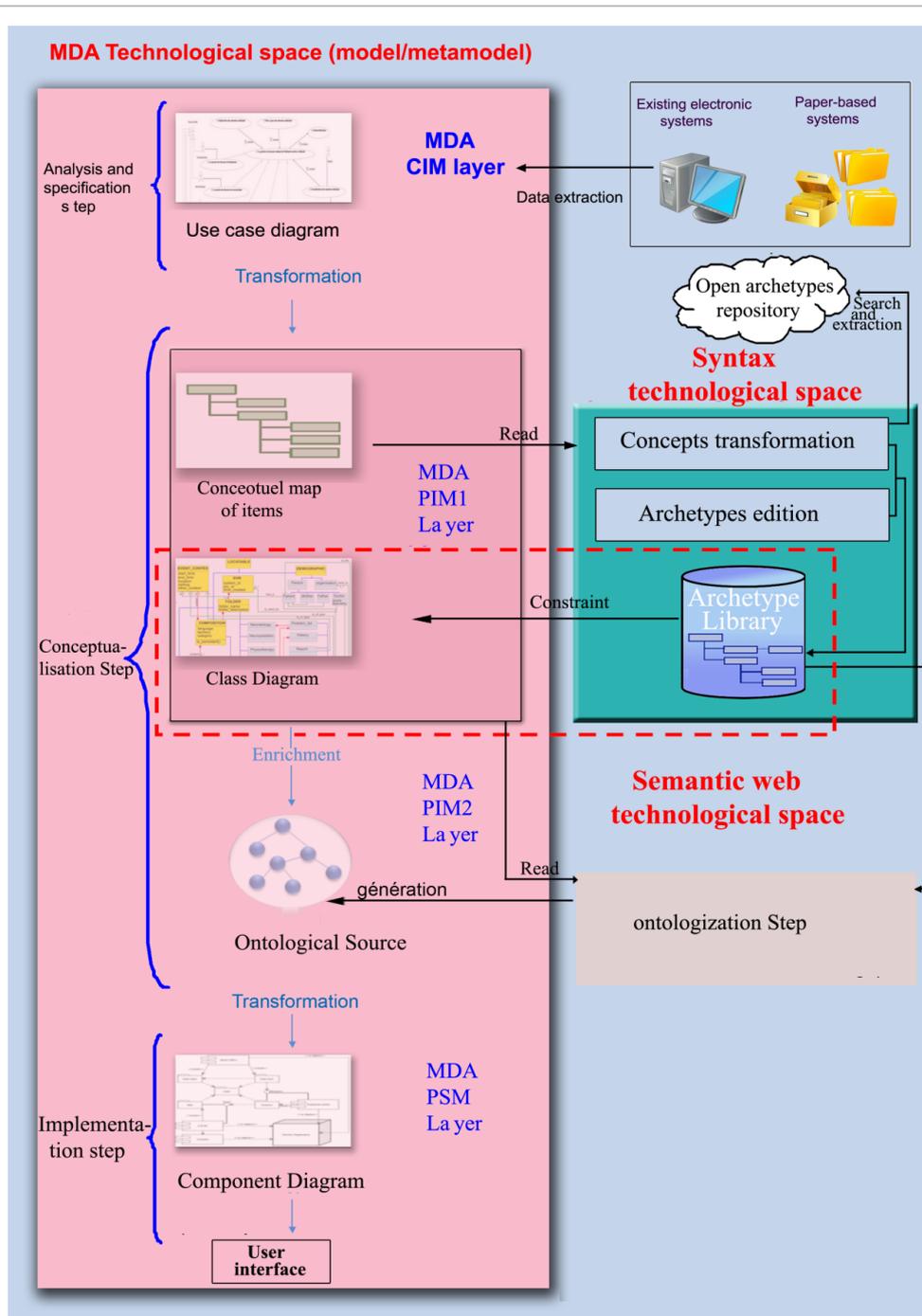


Figure 2: Methodological approach of OpenEHR-MM.

- Permissions: They define the conditions for access to medical data.
- Skills: They represent a combination of knowledge on the level of expertise of a health care actor.
- Preferences: They describe the privileges of presentation of medical information, e.g., preferred language and presentation styles.

According to a context parameter or a combination of these parameters, user interface can change appearance and even content. For example, widgets that make up this interface can take many forms, e.g., select list or text field. These widgets can also change the orientation or even style of visibility. Some additional data, which clarifies the meaning of the fields, can also be displayed. Figure 3 provides an overview of the interactions between context parameters, medical data, which are modeled using the archetype approach.

The generation process of context-aware interfaces that we adopt is based on the use of the following three modules:

- Context acquisition module: This module is responsible for capturing context information. The functionalities of this module are performed using an event handler. This last is responsible for detecting the occurrence of predefined events and then using the context processing module.
- Context processing module: This module serves as an interface between the acquisition module and the adaptation module. It is responsible for the retrieval and transformation of the context information received by the acquisition module, analysing it, eventually aggregating it and finally storing it in appropriate structures (e.g., table of values).
- Medical interfaces adaptation module: This module is responsible for the application of the adaptation rules according to the context parameters provided by the previous module. An adaptation rule defines: (i) The conditions for selecting a user interface appropriate to the received context parameters and (ii) The actions to be performed when these conditions are satisfied. These actions consist in accessing the database in order to extract the characteristics of the appropriate user

interface. Indeed, we have modeled these characteristics at the level of the class “UI_TEMPLATE”.

Results

Through our methodology we experiment an innovative project for the early management of children with a cerebral palsy carried out for the paediatric neurology department of Hedi-Chaker University Hospital in Tunisia-Sfax. In this hospital, the care of patients with Cerebral Palsy involves the following specialties: neurology, paediatrics, psychomotricity, occupational therapy, speech therapy and physiotherapy. This project is based on a highly interdisciplinary care concept, which can only be implemented through efficient communication between all care actors involved in patient therapeutic treatment. In fact, CP EHR must be shared among all the specialties mentioned above. In addition, the information in this EHR must meet the following criteria: (i) To be adapted to the context of use and (ii) Understandable by the various specialists involved. Note that we have apply our modelling approach, described in our previous work [22], for the data collected from the different specialties related to the treatment of patient affected with Cerebral Palsy.

In order to experiment the feasibility of openEHR-MM methodology for Cerebral Palsy EHR, we identified the therapists’ point of view for each clinical concept used. Then, we determine the behaviour of the associated archetype at the level of each specialty. We also identified the different archetypes resulting from the transformations of medical concepts for the different specialties. From the found results, we can deduce that some archetypes are visible for some specialties and not for others, e.g., the “history” and “Problem_list” archetypes belonging to the “COMPOSITION” class appear in the CP EHR for the neonatology specialty, whereas they are not present in the case of physiotherapy specialty. Figures 4 and 5 illustrate the archetypes present in the CP EHR, respectively for the specialties neonatology and physiotherapy.

Furthermore, templates allow us to operationalize the archetypes of our EHR system according to the specific needs of the specialties concerned. The elements of the templates associated with the user interfaces correspond to the fields of the different clinical information of patient entry forms. We limit ourselves in this article to present

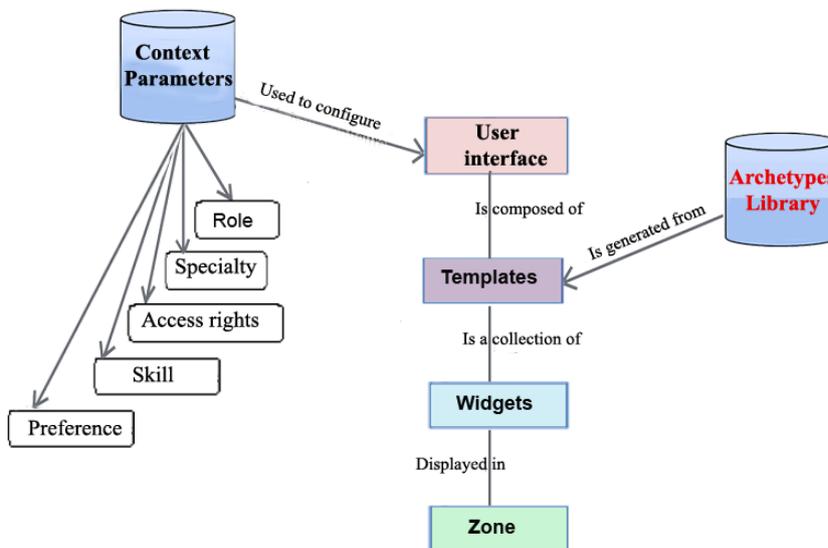
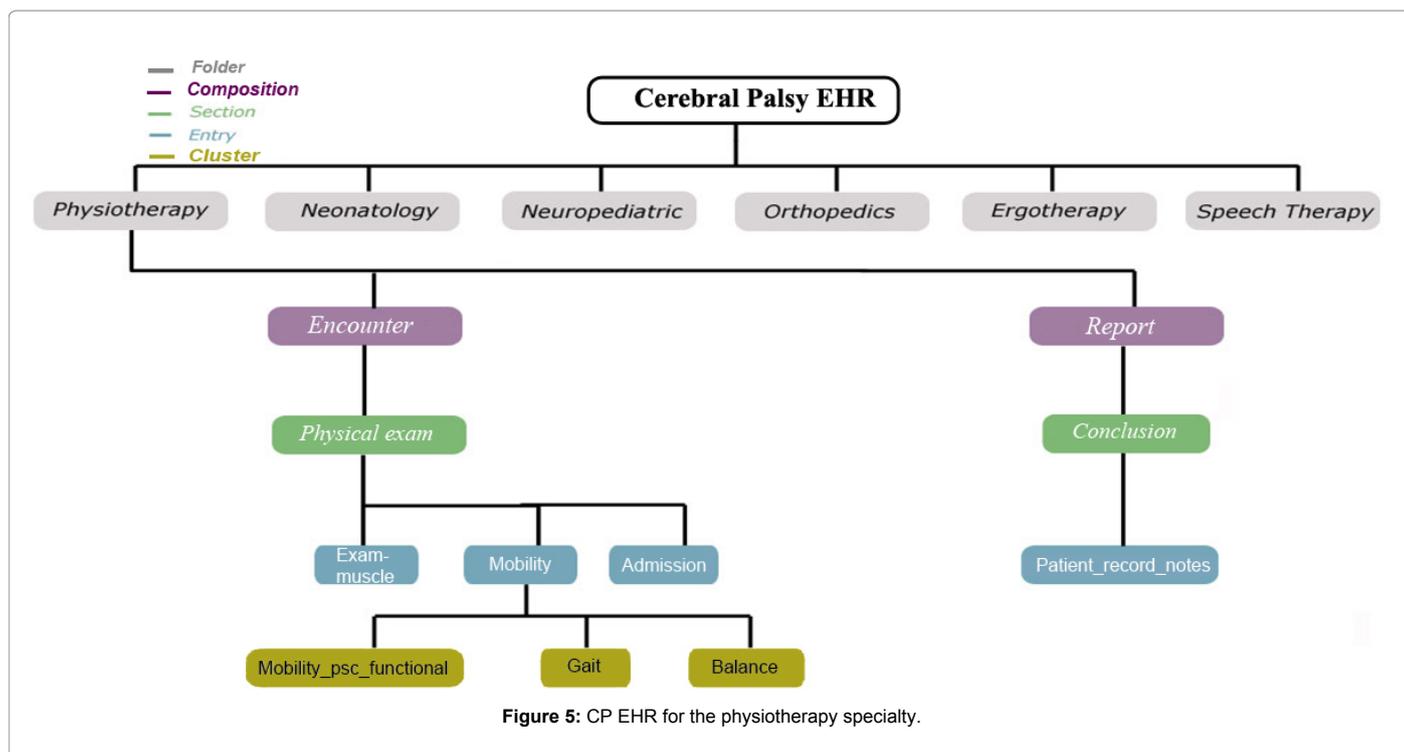
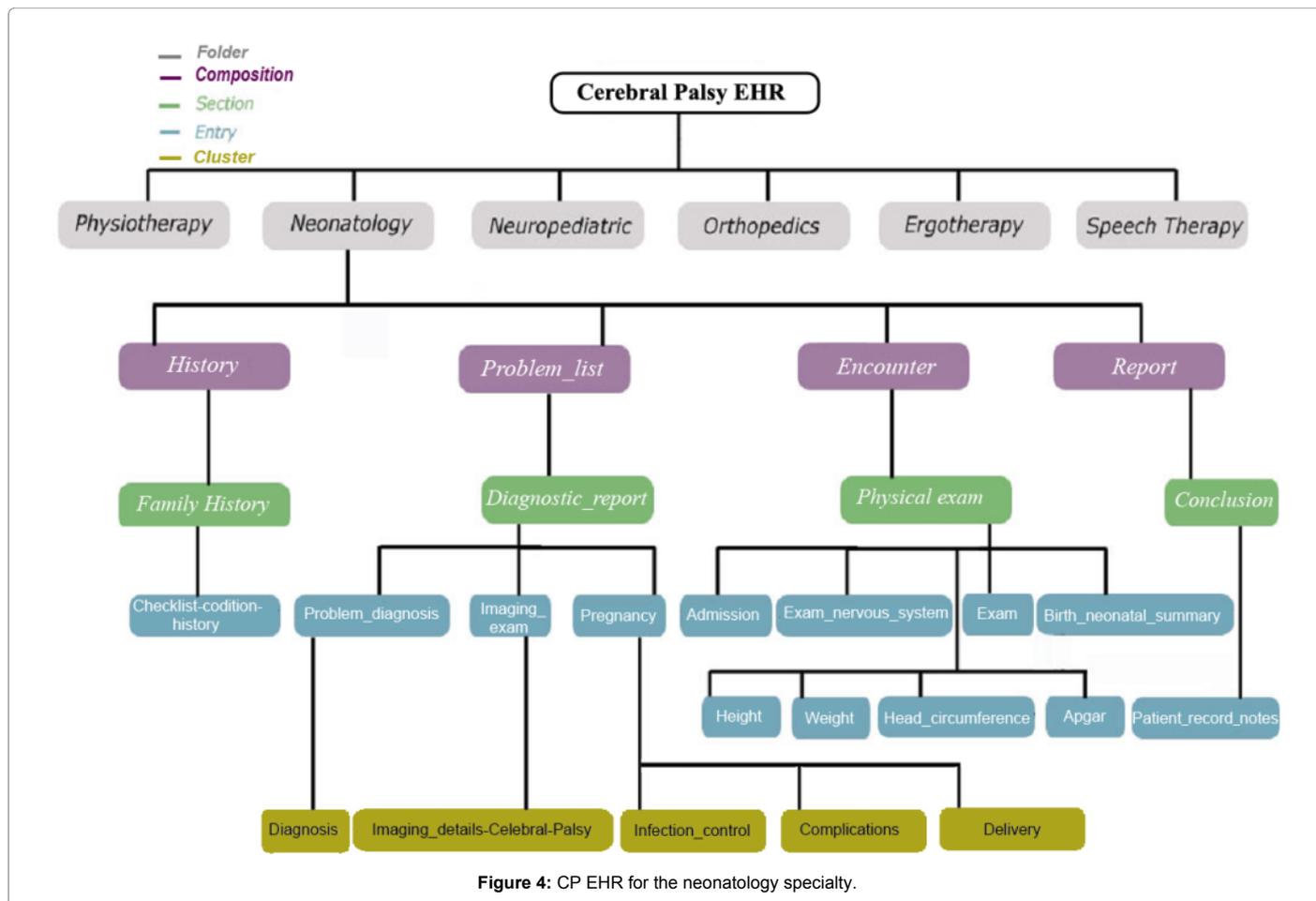


Figure 3: Interaction between context parameters, clinical data and user interface.



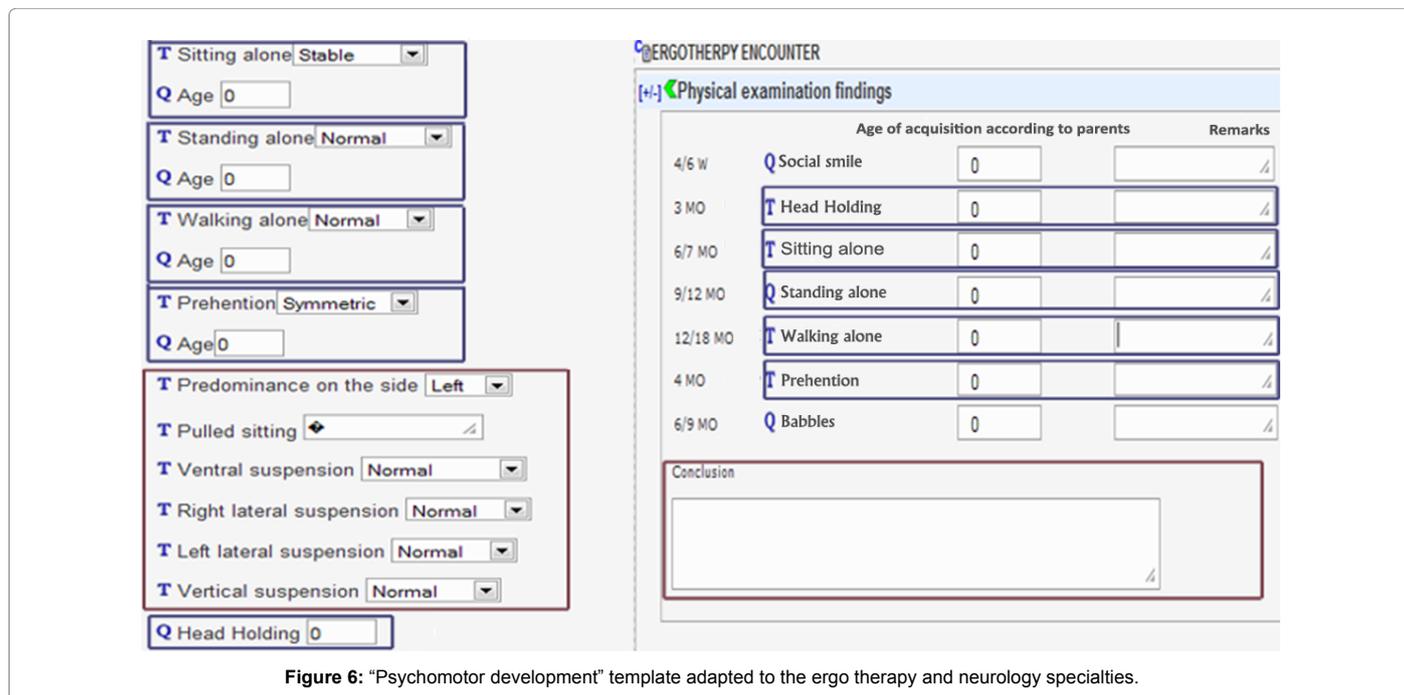


Figure 6: "Psychomotor development" template adapted to the ergo therapy and neurology specialties.

the template "Psychomotor_development", which is shared between the specialties neurology and ergo therapy in order to allow the management of the age of acquisition of the different forms of psychomotor development of a patient. At the level of the specialty neurology, it is also used to allow the management of the data relating to the state of reflex to the movements of the patient. As a result, some widgets, such as "predominance on the side" and "Pulled sitting", are only visible in the specialty neurology. The widgets associated with the "social smile" and "head holding" elements are more important for the occupational therapy specialty than for the specialty neurology. Hence, the orientation of these widgets must change according to the relevant specialty. The "Sitting alone" and "Standing alone" widgets take the form of a selection list for the specialty neurology and the shape of a text field for the occupational therapy specialty. Figure 6 illustrates two extracts associated with the user interface containing the "Psychomotor_development" template adapted according to the considered specialties.

Discussion and Conclusion

In this paper, we propose the OpenEHR-MM methodology up to the generation of context-aware medical interfaces for EHR systems according to the openEHR standard. This methodology is based on the use of the MDA technology space. Indeed, the study of this space allows us to observe that the architecture MDA, which proposes a set of methods to construct and manipulate models, is well suited for the development of our methodology. Indeed, (i) Meta-modelling helps us to divide the modelling process into several levels of abstraction, and (ii) Model transformation has enabled us to guarantee interoperability to or from models related to information levels and knowledge.

The OpenEHR-MM processes are based on open source tools created by the organization "Ocean Informatics" [27], in this case "Ocean Informatics Editor", "Template Designer" as well as The CKM clinical knowledge manager. The experiment of OpenEHR-MM was carried out on the case of children with Cerebral Palsy.

The adaptation approach we define is based on the modelling of the context parameters in the "CONTEXT" class belonging to the UML class diagram. Noting that the process of transforming this diagram into an ontological source, which we define, allows us to benefit from the advantages of using ontology. Indeed, this ontological source can be assimilated to a knowledge base containing the concepts relating to context information, the semantic relations between these concepts and the rules of inference of new knowledge. This may involve defining rules governing the behaviour of concepts defining the user interface, such as archetypes and views associated with these archetypes.

We see the need to extend the reference model of openEHR in order to support knowledge related to the presentation and the values of the returned data and thus be able to provide interfaces adaptable to the context of the user. To this end, we have introduced a new concept called "context-aware reference model". This extension is used to model the context parameters within the UML class diagram. At this level, we introduce components and tools for adapting medical interfaces to the contexts of users.

This is because the archetypes modeled using the openEHR standard allow the definition of the structures of the clinical data and do not contain information making it possible to specify how these structures must be adapted to the context of the user. To this end, we make extensions to the information model of the openEHR standard through the modelling of context information.

As part of that work, we are limited to the modelling of the context parameters at the level of the UML class diagram. In our future work, we envisage the transformation of this diagram into an ontological scheme in order to benefit from the added value by the use of ontology. Indeed, this ontological source can be assimilated to a knowledge base containing the concepts relating to context information, the semantic relations between these concepts and the rules of inference of new knowledge. This can deal with the definition of rules governing the behaviour of concepts defining the user interface, such as the archetypes and views associated with these archetypes.

References

1. ISO DTR 20514 (2005) Health informatics-electronic health record: definition, scope and context (1st edn.). *Int Org Stand* 1: 1-27.
2. Martinez-Costa C, Menarguez-Tortosa M, Fernandez-Breis J (2009) A model-driven approach for representing clinical archetypes for Semantic Web environments. *J Biomed Inform* 42: 150-164.
3. Blobel B (2006) Advanced EHR architectures-promises or reality. *Methods Inf Med* 45: 95-101.
4. <http://www.omg.org/spec/AML/Current>
5. http://www.openehr.org/who_is_using_openehr/academic_research. 2016
6. http://www.academia.edu/21215796/openEHR_Architecture_-_Architecture_Overview
7. http://www.openehr.org/downloads/ADLworkbench/working_with_templates
8. <http://www.hl7.org>
9. http://www.iso.org/iso/catalogue_detail.htm?csnumber=40784
10. <http://www.openehr.org/releases/1.0.1/architecture/am/aom.pdf>
11. Chen G, Kotz D (2000) A survey of context-aware mobile computing research.
12. Strang T, Linnhoff-Popien C (2004) A context modeling survey.
13. Barkat O (2017) Utilisation conjointe des ontologies et du contexte pour la conception des systèmes de stockage de données. Autre [cs.OH].
14. Samulowitz M, Michahelles F, Linnhoff-popien C (2001) Capeus: An architecture for context-aware selection and execution of services. *Springer* 70: 23-39.
15. Chaari T, Laforest F, Flory A (2005) Adaptation des applications au contexte en utilisant les services WEB. *Mobilit Ubiquit* 2005: 111-118.
16. Baumgartner N, Retschitzegger W, Schwinger W (2008) Software architecture for ontology-driven situation awareness. *ACM Press* 2008: 2326-2330.
17. Ellouze AS, Bouaziz R, Hanen Integrating semantic dimension into openEHR archetypes for the management of cerebral palsy electronic medical records. *J Biomed Inform* 63: 307-324.
18. Benslimane D, Arara A (2003) The multirepresentation ontologies: a contextual description logics approach. *Seman Scho* 2003: 16-20.
19. <http://www.openehr.org/ckm/>
20. Samet AS, Bouaziz R, Bouassida N (2014) Reusing openEHR archetypes for the expression of cerebral palsy electronic medical records. *Comput Sci Appl* 1: 179-188.
21. http://www.openehr.org/releases/1.0.2/architecture/rm/ehr_im.pdf
22. Achiri S (2012) Une architecture agents pour l'adaptation au contexte des systèmes d'information ubiquitaires. Tech Rep.
23. Mohapatra DP, Patnaik S (2013) Intelligent computing, networking, and informatics.
24. Perera C, Zaslavsky A, Christen P, Georgakopoulos D (2014) Context aware computing for the internet of things: A survey. *IEEE* 16: 414-454.
25. <http://www.oceaninformatics.com>
26. <http://www.ihtsdo.org/>
27. Schilit B, Theimer M, Welch B (1993) Customizing mobile applications. *USENIX* 25: 129-138.