A Service for ClinicSpace Architecture to Provide Context Data Persistence and Context-Based Selection of Documents

Vinícius Maran[1], Iara Augustin[2], Deise de Brum Saccol[2]

[1] DCEEng – Universidade Regional do Noroeste do Estado do Rio Grande do Sul (UNIJUI) - Rod RS 344 – Santa Rosa – RS – Brasil

[2] Centro de Tecnologia – Universidade Federal de Santa Maria (UFSM) – Av. Roraima, 1000 – Santa Maria – RS - Brasil

vinicius.maran@unijui.edu.br, {august, deise}@inf.ufsm.br

Abstract. *Actual healthcare systems suffer from a high rate of rejection by the physicians who use these systems, because it is necessary that the users explicitly provide information constantly to these systems. So, one of the biggest challenges for pervasive healthcare systems is to find how use the environment context information in a simple and functional way between different computer systems. In the literature, ontologies are frequently used for the representation of context and have an important role in pervasive systems if used together with persistence and retrieval of context information. For proposing a solution to the mentioned problems, is under development an architecture called ClinicSpace which focuses on providing assistance to physicians in performing their daily tasks, using concepts defined in ubiquitous computing, which allows the system adapt itself constantly to the user and their needs. This work describes the process of developing an integrated service to ClinicSpace architecture, which supports the use of context data and clinical documents in a distributed manner, and also allows the contextualized selection of clinical documents, using data from the context at the moment of clinical information query.*

1. Introduction

Ubiquitous computing defines a space where computers and digital devices of all kinds are fully integrated with user and his environment. It has as main objective to help the user to perform their daily activities, abstracting the computational issues involved in these processes [Weiser, 1991].

In healthcare domain, ubiquitous systems has as its main characteristic the constant monitoring of the environment and people who are embedded in it; thus, these systems create smart environments which are reactive and proactive [Ferreira et al., 2009]. To facilitate the use of ubiquitous healthcare systems by the users (physicians or patients) and decrease the rejection rate of this type of system in the practice, it is necessary to customize the information, varying it according to the context information.

Currently, there is a lack of tools in healthcare environments that provide abilities to customize the information by clinical professionals and to provide access to relevant information. Therefore, tools are needed to: a) enable the persistence and recovery of context data, both in a simple way - just like a query of context elements;

and (b) provide more complex ways of querying, adapting the execution code and clinical content as the informed context of a particular task in execution.

In order to experience with ubiquitous solutions in healthcare systems and hospital environments, the Gmob group is prototyping software architecture, developed as a set of services, called *ClinicSpace* [Ferreira et al., 2010]. Such architecture aims to adapt electronic health systems to become it personalized by physician. This paper addresses a service that, inserted into the *ClinicSpace* architecture, provides a solution for persistence and recovery of context elements. Such service also offers the ability to persist and retrieve documents and examinations based on the context information provided at query execution time.

This paper is organized as follow. Section 2 presents an overview of ubiquitous systems and their importance for the problem resolution found in current healthcare systems. Section 3 briefly describes the *ClinicSpace* project. Section 4 explains the service that provides persistence and recovery of context data and context-based selection of documents in the *ClinicSpace* architecture. Section 5 shows the performed tests with prototypes in a controlled environment, simulating a real-world use of the system. Section 6 briefly compares the related work. Section 7 concludes this paper.

2. Ubiquitous and Context-Aware Computing

The idea that computing is user-centric and exists in an invisible way to the end users is the features of the Ubiquitous Computing proposal, defined by Mark Weiser (1991). It defines a world where different and varying types of sensors make constant readings on every environmental change in the ubiquitous environment and the core system reacts to these changes. The information captured from the environment enables ubiquitous systems to interact and take action in accordance with user's needs. In this way, applications should be aware of the context, adapting itself without direct user interference. Today, direct user interference usually occurs on the operation of healthcare systems [Ferreira et al., 2009], which can contribute for the high rejection rate for this type of system by clinical professionals [Ferreira et al., 2010].

According to Dey et al (2000), context is defined basically as a range of information that can be used with the objective to characterize the situation of a group of entities. As context data can be represented in many ways, Strang & Popien (2005) performed a comparison between forms to represent context data and the requirements that these representation ways are used for.

This research compared the forms of context representation based on six main features, namely: distributed composition (*dc*), partial validation (*pv*), wealth and quality of the information (*qua*), incomplete and ambiguous (*inc*), level of formality (*for*) and applicability to existing environments (*app*) [Strang; Popien, 2005]; we have added the interoperability factor (*int*) to this comparison. Other investigations [Nicklas et al., 2008] propose hybrid solutions for context representation with the use of ontologies.

However, the researchers do not present solutions for the context element persistence in a distributed way, in order to enable context data access to nodes of a ubiquitous network.

The results of these comparisons are presented in Table 1.

Table 1. Comparison between context representation forms [Strang;Popien, 2005]

Forms of Context	***dc***	***pv***	***qua***	***inc***	***for***	***app***	***int***
Key Value	-	-	-	-	-	*	*
Mark Schema	*	*	-	-	*	*	*
Object Oriented	*	*	*	*	*	*	-
Logic Based	*	-	-	-	*	-	*
Graphic	-	-	*	-	*	*	*
Ontology Based	*	*	*	*	*	*	*

We consider the model based on ontologies as the most suitable for the context representation because it matches the main comparison requirements. In addition, ontologies facilitate the interoperability between systems, a desirable feature of ubiquitous systems. Therefore, this model was chosen to represent context data in the *ClinicSpace* architecture. The *ClinicSpace* project focuses on assistance to clinical professionals during the execution of their daily tasks in a hospital environment. Such project uses technologies from ubiquitous computing in a way that seeks to take a low intrusion into the clinical performance of daily activities.

3. The ClinicSpace Architecture

Electronic Health Systems (*EHS*) are designed to attend the requirements for hospital administrative control. We consider that this approach causes a high rate of use rejection by clinical users. This rejection factor implies on troubles in performing daily activities, especially related to information feed process by clinical users [Bardram et al, 2007]. To oppose this rejection factor, the *EHS* systems should be designed with a clinical-centered view, seeking to help users on their daily tasks.

This view is explored in the *ClinicSpace* project, which allows clinical professionals to customize the system by modeling their daily activities and providing computational support for their execution [Ferreira et al, 2009]. The *ClinicSpace* architecture provides computational resources to help clinical users during the execution of their daily tasks with minimal need of user intervention. In order to allow this aid occurs successfully, the ClinicSpace architecture modeling is based on the Activity Theory [Ferreira et al, 2009] and on the model presented by Ranganathan (2005). To provide the computational support for assistance in performing tasks, the *ClinicSpace* architecture is composed of layers that are managed by *EXEHDA* middleware [Yamin; Augustin, 2005].

Figure 1 shows the architecture separated by levels, namely: (i) *Higher lever*, responsible for the system interface interaction with users - basically composed by the *IETC* (Editing Tasks and Context Interface), it allows users to model and follow their activities status; (ii) *Intermediate level*, composed by *ClinicSpace*'s services, it is responsible for the task modeling and interconnection between the top layer (interface layer) and the bottom layer; and, (iii) *Lower level*, composed by *EXEHDA* middleware, which performs the allocation control of the resources according to the system needs [Ferreira et al, 2009].

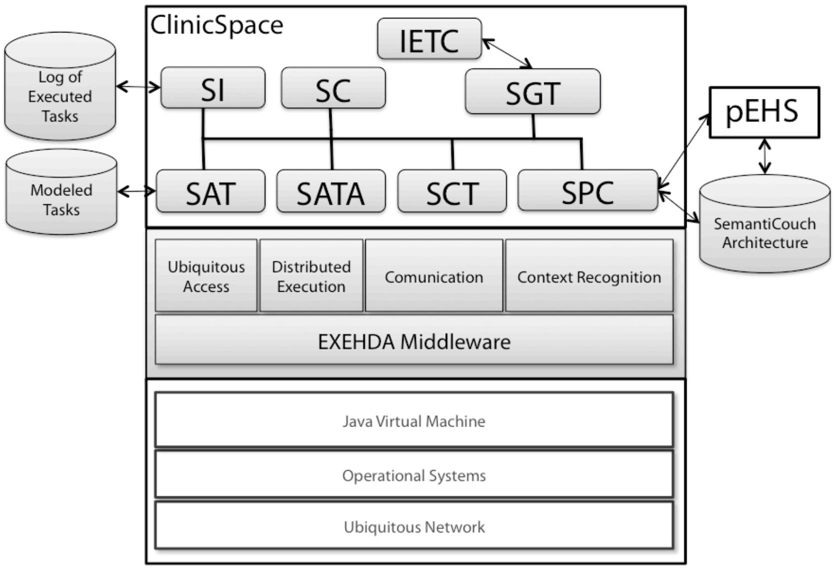

Figure 1. ClinicSpace Architecture

The *ClinicSpace* architecture is divided into services, which perform specific tasks. The communication varies according to the activities performed by users. The architecture consists of the following components: (i) *IETC* (Editing Tasks and Context Interface) (Figure 2), (ii) *SGT* (Distributed Task Manager Subsystem); (iii) *SC* (Context Service); (iv) *SI* (Inference Service); (v) *pEHS* (Pervasive Electronic Health System); (vi) *EXEHDA* middleware [10]; *CouchDB* database [Anderson et al, 2010]; and (vii) *SPC* (Service for context persistence), which is the focus of this paper.

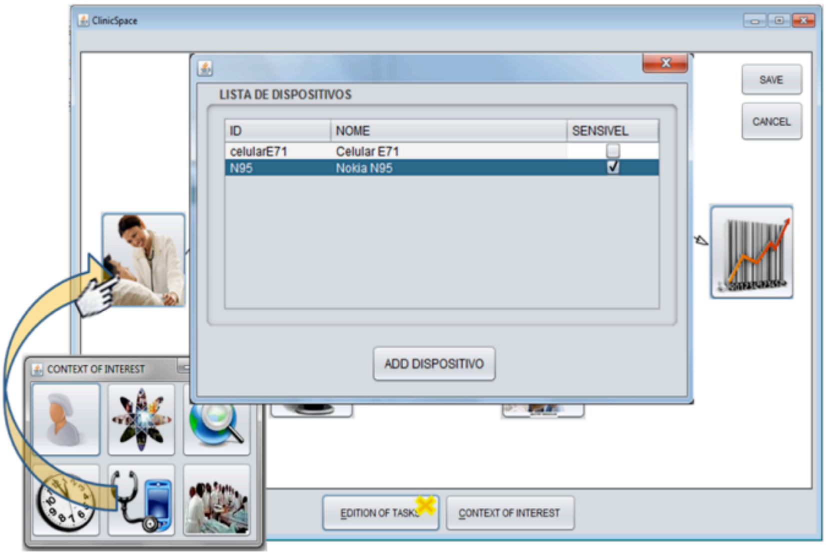

Figure 2. *IETC* user interface [Ferreira et al, 2010]

Since the working model of the *ClinicSpace* architecture has as it main focus to provide help for clinic professionals in their daily activities, it is necessary to create a service that easily allows the clinical document persistence and their recovery, according to the context of the performed activities by the clinical users at query. The next section describes the definition process of the service for context persistence and context-based selection of documents for *ClinicSpace* architecture.

4. Context-Based Persistence and Context-Based Selection of Documents

We propose a new service that allows context data storage and recovery by services in *ClinicSpace* architecture. In addition, the service provides access to stored documents in a customized way, selecting versions according to the context information reported in the query of these documents compared with the context information that are integrated into the document.

One of the necessary features for the context-based selection of information is the context data storage where the documents were generated. The terms in the domain necessary for the representation of context are formalized in the ontology – through the use of OWL files, used as a general description of the ubiquitous hospital environment.

4.1. Context Representation Based on Ontologies

For documents and examinations adaptation in *ClinicSpace* architecture, the ontology originally defined by Ferreira et al (2010) was extended and modified to represent documents and clinical examinations in addition to contextual elements. Thus, the information contained in the documents has semantics and can be easily exported and imported from other formats, such as *CCR* or *HL7* standards. To represent context elements in a computable model, we modeled an ontology that represent context elements that can help choosing a document version to be showed to the clinical user. An overview of ontology classes is presented in Figure 3.

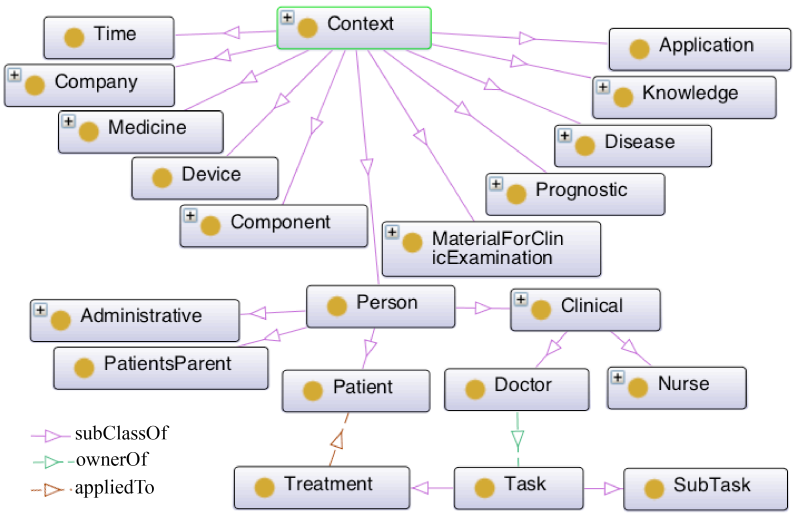

Figure 3. Ontology to represent context elements and clinical documents

In addition to the classes shown in Figure 3, other subclasses and properties were defined, linking classes each other or connecting it to data types and property types. The elements of context defined in this ontology were chosen based on the attributes that can interfere with the adaptation of exams and hospital documents that can be accessed by users in different environmental conditions of use. In addition, documents and exams are modeled and stored within the ontology by the service, allowing the context-based selection of them according to the query context.

4.2. Modeling of Documents and Clinical Examinations

The clinical data was modeled in two super classes: (i) *DocumentPart* class, which defines a small part of a document (subset), defining the information type associated with this part. Its information and the contextual elements will allow the adaptation of this document part; and, (ii) *Document* class, which defines a set of document parts and elements of context that allow the adaptation of the documents in a general form.

This approach was chosen considering the use conditions. For example, if a particular patient performs a standard blood test, it may contain information of interest from different medical specialties, so the context-based selection service can only select parts or versions of documents conforming to the needs of the physician's specialty informed in a particular query. Figure 4 shows the main classes related to the modeling of clinical documents in the ontology.

From the definition of representation and construction form of the ontology that represents context elements and clinical documents, we defined a module for persistence and recovery of context. Thus, the defined ontology could be stored and retrieved in a distributed way between nodes in a pervasive network, enabling access to this information anytime and anywhere. Such module is described in the next section.

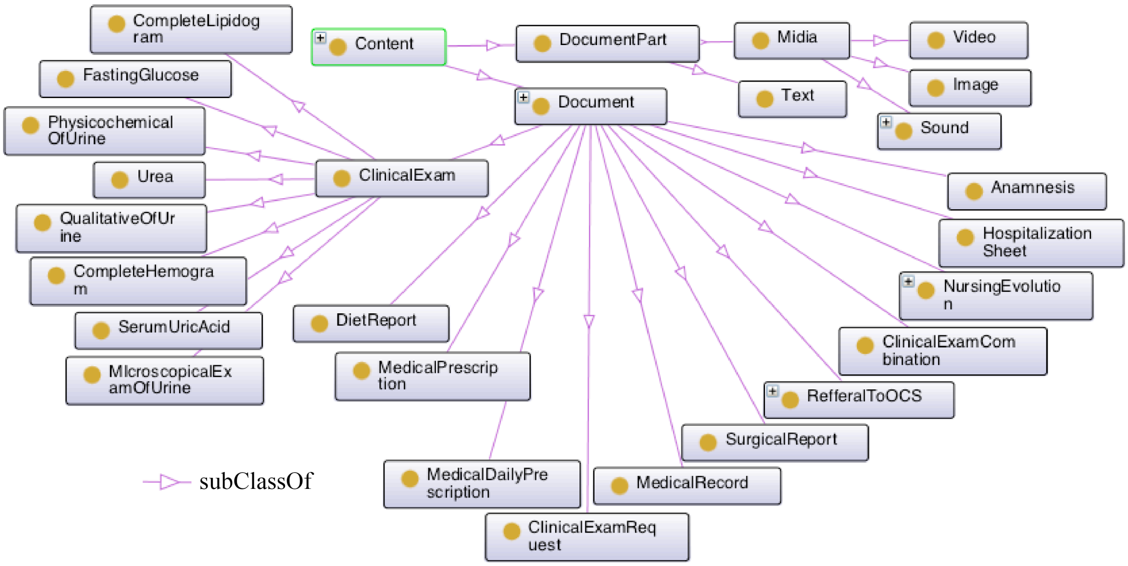

Figura 4. Document modeling in the ontology

4.3. Storing and Recovering Context Data

In order to provide interfaces to the storage and recovery of context and offer support for ontology distribution, we defined a persistence module for ontologies. Such module allows storing context data as an integrated service inside the *ClinicSpace* architecture, which permits access to ontologies that represent the clinical context.

As the core for the storage and recovery of ontologies, we created a solution to convert (importing and exporting) the structure and information of OWL files into the *CouchDB* database. Through the easy replication of JSON documents into the database, we provide the access to the data stored in the database in a distributed way between nodes of a network - be it fixed or mobile. Figure 5 shows a basic representation of the ontology persistence module. This module aims to offer a tool that allows the use of the

CouchDB database distributed in many nodes – in this case, in many hospital institutions, making the information accessible anywhere and anytime. With ontologies, it provides tools for conversion and use of OWL entities in Java language (through the use of a Java objects set defined in the API). We use the *CouchDB* database due some key requirements: (i) Easily scalable: the automatic replication of documents between *CouchDB* nodes makes it scalable in a simple way; this is an key feature because of the amount of data generated by systems based on context information; and, (ii) Based on documents: the use of JSON documents by *CouchDB* facilitates the conversion process of ontologies to a database schema and the retrieval of database data.

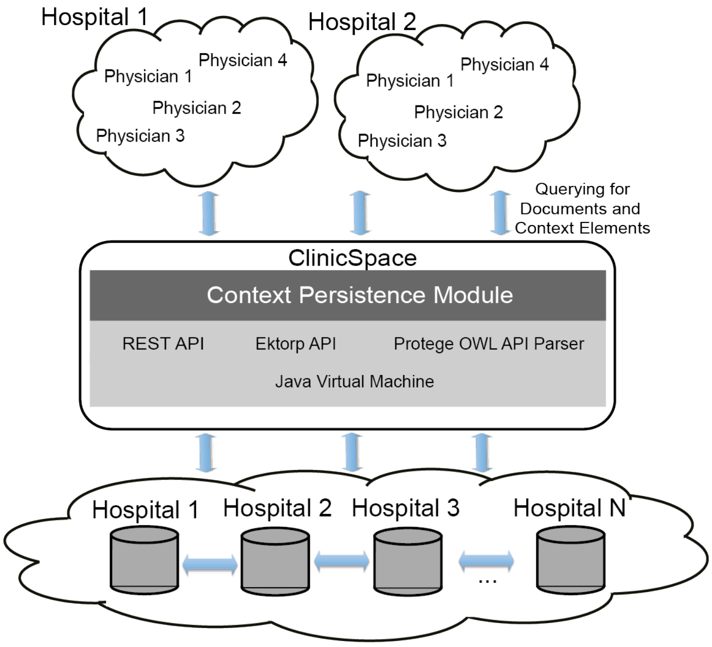

Figure 5. Basic representation of ontology persistence module

The Context Persistence Module works as a top-level layer above the instances of *CouchDB* database. It controls the database nodes and the performed processes, providing a set of classes to create an easy link between the Java programming, the ontology structure and the persisted documents into the database.

The context persistence service works by performing version control of documents. Whenever a document is inserted into the architecture by the *ClinicSpace* services, it is also inserted some context information related to the insertion or modification of this clinical document. For example, when a blood test is inserted into the system, information about the physician, medical specialty of each part of the exam, possible versions of viewing by different types of devices, patient information, among others, are inserted with the documents. In this way, the service selects the documents to be returned by the context data of the environment informed with the query, and after this process, the service compares them with the context information stored with the documents in the ontology.

4.4. Context-Based Document Selection

The first step in the contextualized selection of documents occurs at the insertion of these documents in the system (Figure 6). When the document is inserted into the system, some elements of context – as the medical specialty involved in every part of the document – are added explicitly. Other information is directly associated with the environment in *ClinicSpace* – as the location where the examination was performed and the format of information represented in the document.

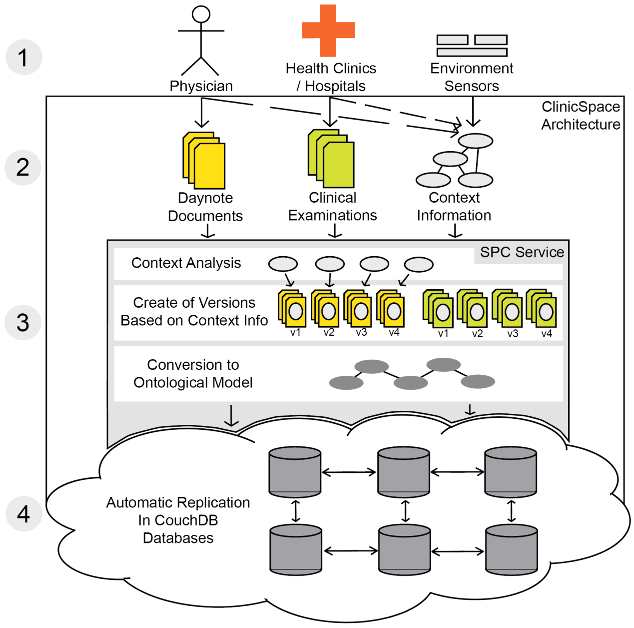

Figure 6. Document inclusion process into *ClinicSpace* architecture

The first step of the process consists in constantly monitoring of clinicians activities by environmental sensors in the hospital environment (i). These processes and activities generate clinic information, which are handled by *ClinicSpace* architecture – be it clinical or context information (ii). After this step, the architecture makes an analysis of clinical and context data (iii), creating versions of these documents according to context; then, it creates ontological instances of this information to persist it in the healthcare system. To provide pervasive access to this information – anywhere and anytime - the system replicates this information between nodes distributed among hospitals (iv).

After inserting the document into the system, the service combines elements of this document to the context information and stores this information into the ontology, in order to automatically create new versions for specific variations of context. For example, versions for different display formats, when the document parts are based on image files. To make the document selection, the service receives information from the context of the query, provided by *ClinicSpace*'s services, and which document will be queried. From this information, the service chooses which version (among versions that were automatically generated) fits the elements of informed context.

For querying, the module uses a group of defined methods that are converted to *JavaScript* scripts to query the *CouchDB* database. They keep document versions automatically replicated between nodes of a distributed network. in our scenario, the nodes of an ubiquitous network can be devices used by physicians or healthcare institutions. For retrieving the best version of the information requested by the user, the system makes a query fetching data from the nodes that contain clinical data that matches the query keys informed by the user (i). After this step, the system performs a comparison between the context provided at query execution time and the context data presented in the version of each document (ii). If a specific version of the context is not found, the system returns another version closed to the ideal – through the comparison of the ontology classes of the individuals. After performing these tasks, the system returns the best version – based on context - to the clinical user (iii).

To demonstrate the features so far prototyped, we next describe a possible usage scenario of the architecture in a controlled environment.

5. Usage Scenario

We consider the following situation: the *ClinicSpace* architecture is used in a hospital on the computers of medical professional as well as on their mobile devices. The hospital has a ubiquitous network, through the use of the ontology persistence module based on *CouchDB* database, where all the exams of all patients are persisted into the database.

While performing the task of patient care, a physician wants to query the information from neurological examinations performed by a particular patient that is being served by this physician at the data consulting time. Despite having a full sheet on the system, with all examinations of the patient that is being treated, the physician needs only to: (i) Obtain the information related to neurological examinations focusing on the patient that is being served at that time; and, (ii) Obtain the examinations in a adapted way to display on your cell phone (phone which is querying the patient information).

5.1. Populating the Ontology and Storing the Information

To represent the usage scenario, we populated an ontology used to represent documents and context elements with information about fictitious physicians, patients and examinations performed by these patients. After populating the ontology, we inserted it into the system by using the interface to convert ontologies into the system format - using JSON documents and storing it in *CouchDB* database. In addition to the inclusion of patient and context data, we created medical examinations with fictitious images for insertion into the database (Figure 7).

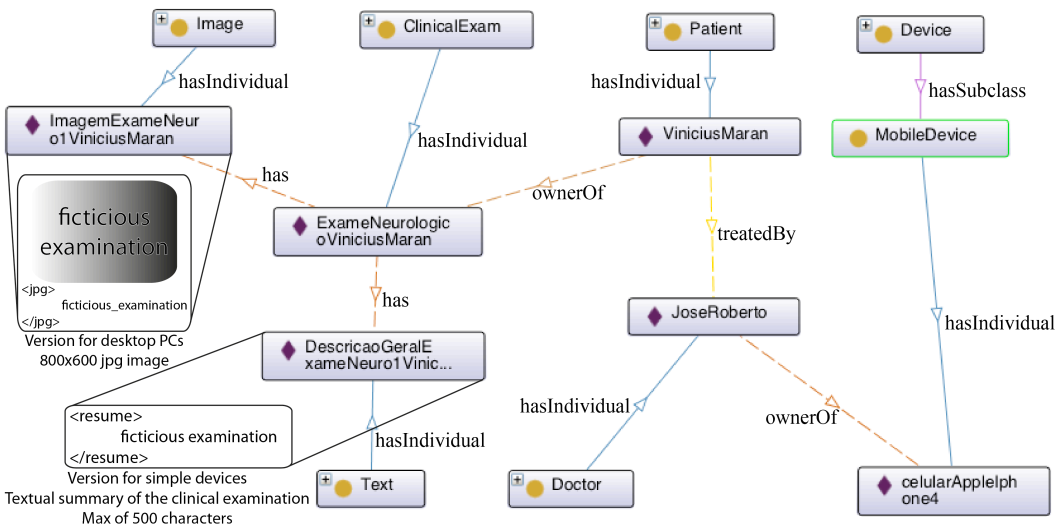

Figure 7. Individuals created to represent the usage scenario

To prove the usage scenario in a controlled and distributed environment, we created three local networks – representing different hospital institutions - with CouchDB databases interconnected by the *EXEHDA* middleware. Thus, we performed ontology insertions and queries that represent fictitious clinical and context data, checking the results of replication and query between nodes of these networks.

We compared the loading time of required applications to implement the tasks in the architecture ClinicSpace using the context persistence and with the last solution used in ClinicSpace architecture [3]. In this test were created instances of physicians and these were gradually inserted into the architecture. The result is mapped in Figure 8.

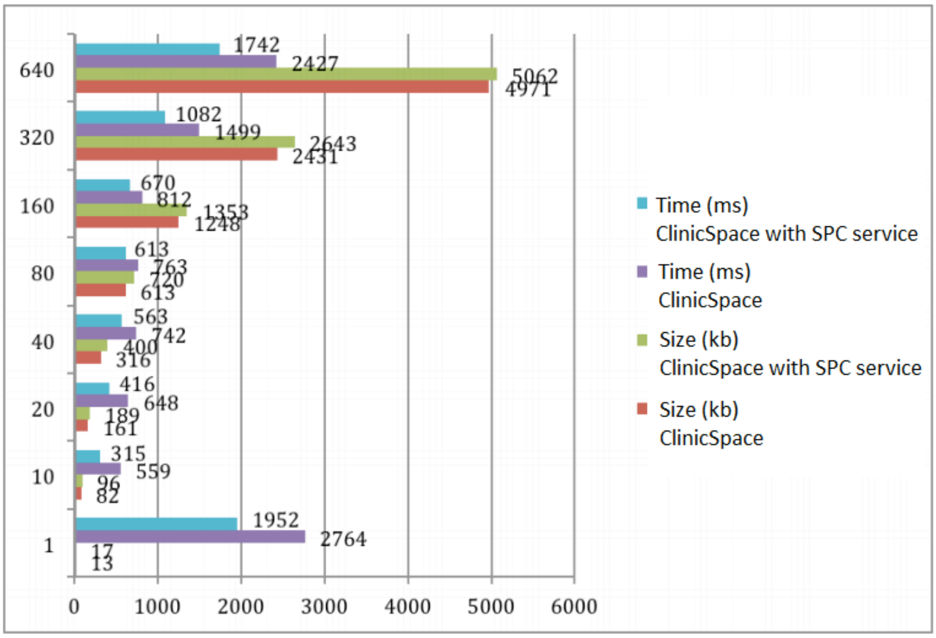

Figure 8. Comparison between the time of loading the system with performed changes

As can be observed, the loading time of system reduced approximately 28% with the insertion of the proposed service in this study. This change is mainly due by the persistence of the ontology data in a database. The loading time with a single physician is higher because at this stage the services of EXEHDA middleware are initialized. After the loading of EXEHDA middleware, instances of physicians were being added gradually. Therefore, (i) the data need not be loaded into memory each time the system is started, and (ii) the data in context and patients' clinical data can be used in a distributed way and accessible in a variety of devices, keeping synchronized automatically. After the performed tests, we have done a comparison between the proposed architecture of this paper and other works, as next discussed.

6. Related Work

The computing based on assistance for the execution of tasks has as first proposal the Aura project [Sousa; Garlan, 2002]. The main feature of this project is that it operates as a pro-active middleware, i.e., the user does not make settings for the system to work according to your needs. The Gaia project [Roman et al, 2003] defines a pervasive space; it is programmable and interactive with the user, who interacts with objects and environments that provide information and resources.

Specifically on the content adaptation issue, the project presented by Dolog et al (2003) defines a form of representing information based on RDF language, allowing them to adapt as created profiles in the same language. The SECAS project [Chaari et al, 2007] presents a platform for adaptation of documents based on RDF language, adapting documents according to contextual elements there also described in RDF language. Table 2 shows a comparative table between healthcare systems and the *ClinicSpace* project.

Table 2. Healthcare project comparison

	Mobility	Context	Automation	Customization	Control
Gaia	X	X	-	-	-
ISAM	X	X	-	-	-
Aura	X	X	X	-	X
ClinicSpace with SPC service	X	X	X	X	X

As we can see, these proposals do not have a vision focused on the end user (healthcare professionals) proposed by *ClinicSpace* architecture. In addition, these projects do not use the resources of representation model based on ontologies for representing the stored information and the contextual elements that affect the adaptation of clinical documents, disabling the use of semantics and tools for inference process of these data.

7. Conclusions

The use of ubiquitous systems in the healthcare field makes the hospital more intelligent and focused on clinical professionals. The proposal of ClinicSpace project aims to reduce the rejection factor currently found in hospital systems by clinical professionals. For that, it provides an infrastructure for applications and services that can assist physicians to perform their daily tasks in a hospital environment in a personalized way.

The context persistence service, include in ClinicSpace architecture, is proposed as means to provide the storage and retrieval of documents, assisting in the process of adapting applications and documents according to the context, and providing them to the users in a individualized way.

The next steps of this work are: (i) to establish procedures for creating document versions in an automated form, automatically converting formats and languages and, (ii) to expand the adaptation forms, inserting new context elements that may help in document adaptation and (iii) to test adaptation with clinical professionals in a real-world environment in order to verify the possible improvements related to document adaptation and presentation.

References

Weiser, Mark. “The Computer of the 21st Century”. Scientific Americam v.265, (1991) Ferreira, G., Librelotto, G., Silva, L., Yamin, A., “Middleware for management of end-user programming of clinical activities in a pervasive environment” In Workshop on Middleware for Ubiquitous and Pervasive Systems, Vol.389, pp.07–12. (2009)

Ferreira, G., Augustin, I; Librelotto, G; Lorenzi, F; Alencar, M; Extending a Middleware for Pervasive Computing to Programmable Task Management in an Environment of Personalized Clinical; In: UBICOMM, 2010

Dey, A.K. Abowd, G.D. (2000) "Towards a Better Understanding of Context and Context-Awareness". CHI 2000 Workshop on the What, Who, Where, When, and How of Context-Awareness

Strang, T., Popien, C. (2005) “A Context modelling survey”. In Proc. Of the Workshop on Advanced Context Modelling, Reasoning and Management as Part of UbiComp.

Nicklas, D., Matthias Grossmann, Jorge Manguez, Matthias Wieland, "Adding High-level Reasoning to Efficient Low-level Context Management: A Hybrid Approach," percom, pp.447-452, 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications, 2008.

Bardram, Jakob E. and Christensen, Henrik B. Pervasive Computing Support for Hospitals: An overview of the Activity-Based Computing Project. IEEE Pervasive Computing, vol. 6, issue 1, p. 44-51, (2007)

Ranganathan, A. et al (2005) “Olympus: a high-level programming model for pervasive computing environments”, in: 3rd IEEE International Conference on pervasive computing and communications, Percom, p. 7-16, 2005.

Yamin, A. and Augustin, I. (2005) “EXEHDA: Adaptative middleware for building a pervasive grid environment” Frontiers in Artificial Intelligence and Applications – Self – Organization and Autonomic Informatics (i), volume 135. S.l.: IOS Press.

Anderson, J. C., Lehnardt, J., Slater, N. (2010 “CouchDB: The Definitive Guide”. Livro. Ed. O’Reilly Media. ISBN: 978-0-596-15589-6. 272 pag. 2010.

Sousa, J. and Garlan, D. “Aura: An architectural framework for user mobility in ubiquitous computing environments” , The Working IEEE/IFIP Conference on Software Architecture (WICSA) (2002)

Roman M, Hess C., Cerqueira R., Campbell and Nahrstedt K. “Gaia: a Middleware Infrastructure to Enable Active Spaces”. IEEE Pervasive Computing. 74-83. (2003)

Dolog, P., Gavriloaie, D., Nejdl, W, Brase, J., “Integrating Adaptative Hypermedia Techniques and Open RDF-based Environments” In: Proc. Of 12th International World Wide Web Conference. (2003).

Chaari, T., Laforest, F., Celentano, A., “Adaptation in Context-Aware Pervasive Information Systems: The SECAS Project”, In: Journal on Pervasive Computng and Communications (IJPCC). Pg. 400-425, Emerald Group Publishing Limited. 2007.