Workflow Model for Management of Ontology of Homecare Pervasive Systems

Leandro O. Freitas[1], Ederson Bastiani[2] and Giovani R. Librelotto[3]
[1]Polytechnic School, Federal University of Santa Maria, Santa Maria, RS, Brazil
[2]Department of Informatics, Federal Institute Farroupilha, Panambi, RS, Brazil
[3]Technology Center, Federal University of Santa Maria, Santa Maria, RS, Brazil

Keywords: Pervasive Computing, Homecare Applications, Ontology, Workflow Model.

Abstract: Pervasive computing applied to healthcare is seen nowadays as an alternative to the overcrowding in hospitals. The implementation proactive applications improve the services offered, helping to speed the treatment of patients. We present in this paper a workflow process model to be used for the development of pervasive systems to homecare. It represents the management of the ontologies used by the systems. We describe the processes that compose the workflow, showing a case study to validate it.

1 INTRODUCTION

The current health paradigm, centred at hospitals and clinics, compromises the capacity and quality of care in the health systems. Every year, the queues in hospitals grow due to a set of factors like, for example, high demand of patients, problems of infrastructure, poor internal communication and lack of professional commitment (Andrade et al., 2009).

One way to minimize the delay on these services is to change the way how patients are treated nowadays. Instead of leading the patient to the hospital to receive the treatment, the clinical services could be taken to the patient's house through the implementation of medical applications creating a homecare scenario. Homecare environments are very dynamics, with frequent changes in its context (Zarghami et al., 2011). For example, the patient can live and enter whenever he wants, walk through the rooms or receive visit of relatives and friends. However, besides being at home, the patient should be under constant monitoring, and any sudden worsening on his health condition should be treated immediately.

Relevant issues about care of elderly people in homecare environments must be highlighted. For example, the caregiver can be someone of the patient's family and without knowledge to assist him correctly. Caregivers may become inattentive due to physical and emotional stress. Thus, he could put in risk the health of the patient, once he may have no control or awareness of his actions and also be unable to ask for help in a critical situation (Lemos et al., 2006).

In this scenario, pervasive computing appears as a solution to solve problems like mobility and dynamicity that characterize a homecare environment, supporting the frequent changes in context. Pervasive computing allows access to information anytime and anywhere, providing quick and secure access to the medical record of patients to physicians through applications even if they are not in their workplaces (Chen, 2004). They can take faster decisions about which treatment apply to a specific patient whose health is worsen.

These kind of systems should use information from the environment. One of the most appropriate ways to represent the knowledge of domains is through ontologies. They can be defined as a formal representation of a contextualization. In (Bastiani et al., 2013) we presented an ontology to represent the existing knowledge in homecare environments and an architecture for the development of pervasive systems focused on homecare treatment. These systems could guarantee that the treatment of patients in their houses would have the same quality of those offered in hospitals.

In this paper we focused on the management of the ontology used by the architecture to develop the pervasive systems. We present the workflow to show how the modules of the architectures behave during the working process of the system. The paper is organized as follows: in section 2 we describe some works that can be related to this. In section 3 we describe the

O. Freitas L., Bastiani E. and R. Librelotto G..
Workflow Model for Management of Ontology of Homecare Pervasive Systems.
DOI: 10.5220/0004896102410248
In *Proceedings of the 16th International Conference on Enterprise Information Systems* (ICEIS-2014), pages 241-248
ISBN: 978-989-758-027-7

ontology to be used by systems developed from the architecture. In section 4 contextualize the architecture developed to homecare pervasive systems. The management of the ontology is described in details in section 5. A case study to validate the workflow is described in 6. And in section 7 we present the conclusions of this work and the future possibilities.

2 RELATED WORKS

In (Gassen et al., 2012), the authors propose a methodology for home care systems, where service orchestrations could be performed by non health specialists. They use ontologies to express the possibilities of modelling and provide the necessary semantic during the processes. With this they intend to personalize the process in the houses of patients, once this kind of environment may have a certain level of differences one from another.

In (Paganelli and Giuli, 2011) the authors describe a configurable and extensible service-oriented framework, aiming to facilitate the development of applications to assist patients with chronic conditions while they stay at home. The framework is composed by an ontology-based context model and a context-aware system and is part of a platform of homecare services aiming the sharing of information, organizing actions to be taken in critical situations.

The work developed by (Evchina et al., 2012) aims to find a easy way to manage a smart home, once this kind of environment may become very complex, with different types of information being requested at the same time by the system. The authors use ontologies to reason upon the knowledge of the domain and use them to assist a context-aware framework developed for smart houses. Although the work is not focused on home health care domains, it can be related to this paper due to the concepts and technologies used, like ubiquitous computing and ontologies.

The works described above can be related to this because all use ontologies as a support mechanism during the design of system for homecare environments. Following we describe the ontology developed for homecare environment.

3 ONTOLOGY APPLIED TO THE ARCHITECTURE

The ontology was developed with Web Ontology Language - OWL (Antoniou and van Harmelen, 2003) (Grau et al., 2008), a recommendation from W3C Consortium[1] for this kind of representation. Its role is to represent the relationships between the entities, the activities that the patient can perform during his treatment and the characteristics of the environment. It could be used by pervasive systems aiming to assist professionals during the treatment of patients. These systems ensure a constant communication between the patient's house and the clinic that provide the service.

Based on information obtained from the medical staff and literature (Thirugnanam et al., 2013), we defined a set of classes related to the monitoring of patients (e.g. symptoms or vital signs). Information about health, such as diseases and symptoms, were mapped according to the International Classification of Diseases - ICD[2]. Thus, using standardized terms and concepts, applications are able to understand the information represented in the ontology. The ontology also represents general classes of the environment, such as mobile devices, sensors and rooms. Figure 1 shows a graph with a part of ontology.

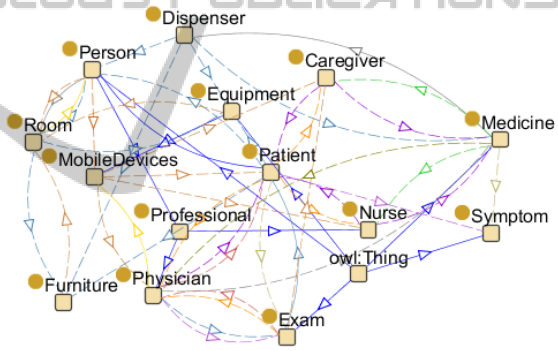

Figure 1: Ontology for homecare environments.

Besides the definition of classes, the ontology also has a set of properties. The datatype properties of a class are the characteristics used to describe them, differentiating its individuals. The class *Disease*, for example, has the datatype property *icd*, which refers the ICD code.

The object properties are used to connect individuals from classes. To illustrate the use of relationships, we can define a scenario where the patient has a symptom (e.g. agitation) and needs some medication. In the ontology, there is a relation between the classes *Symptom* and *Medicine* exists through the object property *combat*, indicating that a drug is indicated for a specific symptom. We also consider that medicines must be in a *Dispenser* (*Medicine isIn Dispenser*). In this scenario, the pervasive system could

[1] http://www.w3.org/
[2] http://www.who.int/classifications/icd/en/

show to the patient or the caregiver, through a screen, where the nearest dispenser that contains the drug is indicated for that situation. The application could be executed every time that the system identify an agitation behavior by the patient.

To to this, we define inference rules using the Semantic Web Rule Language (SWRL) (Horrocks et al., 2004) and queries using the Semantic Query-Enhanced Web Rule Language (SQWRL) (O'Connor and Das, 2009). Considering the previous example, a query could be executed to return the dispenser with the medicine based on its identification code.

```
Medicine(?m)^name(?m,?n)^Dispenser(?disp)^
swrlb:contains(?n, "Name of Medicine") ^
isIn(?m, ?disp) ^isIn(?disp, ?room)
-> sqwrl:select(?disp, ?room)
```

Knowing the local of the medicine, the system performs an inference rule and, based on this rule, sends a notification to the patient:

```
Patient(?p)^isIn(?p,?room)^
Equipment(?equi)^status(?equi,?s)^
sqrwlb:equal(?s,true)^isIn(?equi,?room)^
Notification(?note)
->dispTarget(?note,?equi)
```

After that, the system shows the information about the dispenser and medicine in a device considering the location of the user, which is also mapped in the ontology.

4 ARCHITECTURE FOR HOMECARE PERVASIVE SYSTEMS

In this section we contextualize the architecture for pervasive homecare systems. The system, based on informations captured from the context of the environment, is able to assist the treatment of patients, reducing the caregiver burden and improving the quality of the patient's life. The Figure 2 shows an overview of the architecture of the system.

The system is divided in two main domains that exchange information about the entities involved: the homecare environment and the cloud computing. The homecare domain is composed by six modules. The first refers to the *sensors* which should be distributed in all rooms of the house. Their role is to search for any relevant changes in the context that may impact to the patient's treatment. We will not discuss types of sensors that can be used, since the architecture imposes no restriction on the choice of the devices or how to implement them. In the architecture

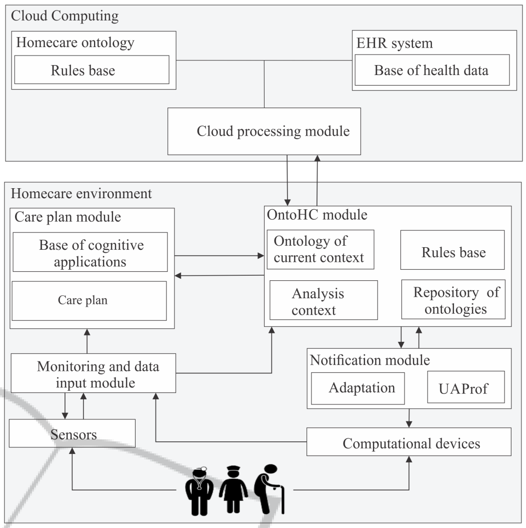

Figure 2: Architecture for pervasive homecare systems.

modelling we considered three classes of variables to be monitored: (i) *environmental* conditions like incidence of light, sound, humidity, temperature, among others, that may influence on the health of the patient; (ii) *physiological* data which includes blood pressure, oxygen saturation level and pulse rate; and (iii) *behavioural* aspects, since the change in the patient's behaviour may indicate change in disease stage or non-compliance with a prescribed treatment.

The *monitoring and input data module* is responsible for the insertion of data into the system. It receives information from the sensors or computational devices and process it standardizing the signals to a specific format. This step is important because a homecare domain may have different types of sensors and devices, with different processing platforms. So, the information must be processed before being sent to the *Care Plan module* or *OntoHC module*.

The *Care Plan module* stores all the prescriptions that the patient must fulfil during treatment (e.g. schedules of medication or exercises), and has a *base of cognitive exercises* that the patient could perform (e.g. a memory stimulation exercise). This information can be inserted into the care plan from any computational device that physician uses to access the system, (e.g. smartphone). For each prescription the system generates an event to check if the patient is in compliance with the treatment. To do that, the system communicates the *OntoHC Module*.

The *OntoHC Module* contains the *Ontology of Current Context*. Its structure will have only instances of classes detected by sensors in a specific scenario, during the treatment of the patient. It was created from another ontology which represents all the knowledge that a homecare environment may have. This module is also responsible for verifying if all instances detected by the sensors are already mapped in the *Ontology of Current Context*. In affirmative case, the module access the *base of rules* to check if there is any query or rule to be performed, and sends a message to the *notification module*. From these queries the system triggers applications to help the tasks of professionals through *computational devices*. To do this, the module considers the device profile using the User Agent Profiling Specification (UAProf)(Alliance, 2001). Each UAProf document provides hardware and software characteristics of a specific device, as capability of receive images and files. With the union of data from the ontology and the capabilities of devices, it is possible customize the interface to final user, considering his preferences and needs.

However, if not all instances of classes detected by the sensors are represented in the ontology, a new one must be created, which will represent the current context and will replace the one that is being used. Then, the old ontology will be stored in the *repository of ontologies*. In this case the *OntoHC module* sends information about these new instances to the *cloud processing module*, located in the cloud computing domain. This module will create a new ontology and its structure will have classes of the instances detected. Then, a search will be performed on the EHR system information about them.

The new ontology will be sent back to the *OntoHC module* and will replace the one that was being used. With the new ontology of current context the *OntoHC module* is able to perform queries and rules seeking for possible actions to be taken that somehow could assist the users (e.g. time of medicine, daily tasks, etc).

More information about the working process of the architecture can be found in (Bastiani et al., 2013). Following we describe how is the workflow of the management of the ontology in the *OntoHC module* and in the cloud computing.

5 MANAGEMENT OF THE HOMECARE ONTOLOGY

The main purpose of this section is to describe the management of the ontology developed to be used with pervasive systems for homecare environments. For this we first describe the *OntoHC module* of the architecture because it is where the creation and inference on the ontology are performed. After that we present the workflow of the management with its validation.

5.1 OntoHC Module

The *OntoHC module* can be seen as the core of the architecture. It has connection with the *Monitoring and data input and Care plan and Notification module*. It also has a connection with the cloud computing allowing it to have access to information store in the EHR system, where the medical history of patients are stored. The *OntoHC module* is composed by a ontology that represents the current context of the homecare environment, a base of inference rules to process the ontology, a analysis of context component and a repository of ontologies already used.

The *ontology of the current context* is composed by only one part of the structure of the homecare ontology described in section 3. This ontology was created after a analysis of the context of a specific moment in the homecare environment and represents only information about the instances of classes that were present in that time. For example, if a physician and patient were detected by sensors in a certain time in a specific room of the house, the system would create an ontology with only the classes *Physician* and *Patient* (and other classes directed related to them) with their relationships. This smaller ontology of current context helps to improve the performance of the system when processing inferences and queries, and also when executing applications.

The *base of rules* is composed by a set of inference rules defined. The rules can modify the structure of the ontology, changing the values of attributes or creating new relations between classes. The queries do not have the power of modifying the structure of the ontology. Their function is to perform searches in the ontology, returning information about the involved when they satisfy the premisses of the query, similar to SQL.

Every time a new context is detected by the sensors, information about the instances involved are sent to the *OntoHC module*. If all of them are already described in the ontology of current context, the system access the base of rules and searches for any rule or query that may be applied to those instances. For example, when the physician arrives the house to his daily visit, the sensors detect his presence and the system performs a query, discovering that there is an exam where the attribute *visualized* (which has range *dateTime*) is blank. In other words, the physician did not analysed the exam yet. When he decides to visualise it, the exam is shown on a screen next to him. When the physician analyses the exam that he had requested, the system detects that the exam is being analysed. When the physician close the application, the system could execute a rule to change the status of visualization of that exam. After that, its value could be setted according the date and time that the physician visualized the exam, characterizing that the physician already performed this task. When a inference rule or query is executed, the system warns the *Notification module* about it, which will be responsible to process this information correctly, starting an application through the *Computational Devices* or starting a new communication with the *OntoHC module*.

The *OntoHC module* also has a *Repository of Ontologies*, which stores all the old ontologies that where used as being of the ”current context” in some point during the flow of the system. When the sensors detects new instances of classes, a new ontology must be created and the system sends the one that was being used to the repository. This repository is important to register all scenarios represented in ontologies and which entities were part of it. With this, it is possible to access old contexts, seeking for information, like, which nurse were on duty in a specific day, how was the vital signs of the patient or if he performed an usual task on that day.

When at least one of the instances detected by the sensors are not mapped on the structure of the ontology of current context, the *OntoHC module* starts a connection with the cloud computing domain and requests a new ontology. It sends information to the *Cloud Processing module* about the instances that will be used to create the new ontology. Also, it sends information about the ontology of current context. This information will be used to update the ERH system and will ensure that when the new ontology is created, the information inserted on it is not out of date.

The last component of the *OntoHC module* is the *Analysis of Context*, which is responsible to analyse the information sent by the *Monitoring and data input module* and process it. Based on the information received, the *Analysis of Context* component searches in the base of rules for any query or inference rule that may be performed for those entities.

5.2 Workflow Process Model

This section presents a workflow model aiming to show how the ontology is manipulated during the execution of the system when it accesses the *OntoHC module* and when it requests a new ontology for the cloud computing. The workflow model was developed with the software *Bizagi Process Modeler*[3]. Figure 3 shows how the ontology is managed during the execution of the system.

[3]http://www.bizagi.com/index.php/en/products/bizagi-process-modeler

In step *a* the *OntoHC module* receives information from the *Monitoring and data input module* and according to the type of information it will be treated differently. If it refers to new instances of classes detected by sensors, that may configure a relevant change in the context of the homecare environment and the flow goes to the step *b*. However, if the information was inserted manually, the system verifies if it refers to a new rule and, in affirmative case, the *Base of rules* is updated (step *n*). The more inference rules it may have, the more proactive the pervasive system may become. If the information placed manually is not a rule (e.g. new daily task, or new medicine posology), the flow should jump to the step *b*.

In step *b* the system verifies if the instances detected are all structured in the *ontology of current context*. To do this, it accesses the *Base of Rules* and executes queries on the ontology using the identification codes of the instances as parameters. If at least one instance is not in the classes of the ontology, the flow goes to step *c*, otherwise it goes to *i*. Step *c* has the duty of sending a requisition to the cloud computing domain with information about the entities that should be mapped in the new *ontology of current context*. With that, information about the current ontology also must be sent to update the EHR system.

The *cloud processing module* (step *d*) receives two sets of data. The first refers to the current information used by the system (e.g. patient’s vital signs or symptoms). This will be inserted into the EHR system ensuring that the next ontologies will have updated information. The other set of data refers to the instances present in a context when a change was detected. This information will be used to *Create a New Ontology* (sub-process of the process *d*).

After the creation of new ontological structure, the system starts a connection with the EHR database asking for information (step *e*), which will be inserted in the ontology in step *f*. Then, the new *ontology of current context* is sent back to the homecare environment with updated information about the instances detected by the sensors (step *g*).

This new ontology will be used as *ontology of current context*. When the system receives it in step *h*, the ontology that was being used before is be sent to the repository of ontologies, in the sub-process *Store the old ontology*. Meanwhile the flow continues to the next stage. The process *i* is executed when a new ontology is sent from the cloud or the result of question in process *b* is ”Yes”. In this step the system verifies, according to the information detected by the sensors, if the there are any activity related to the patient that can be performed. In affirmative case, the flow goes to step *j* where the system starts a connection with the

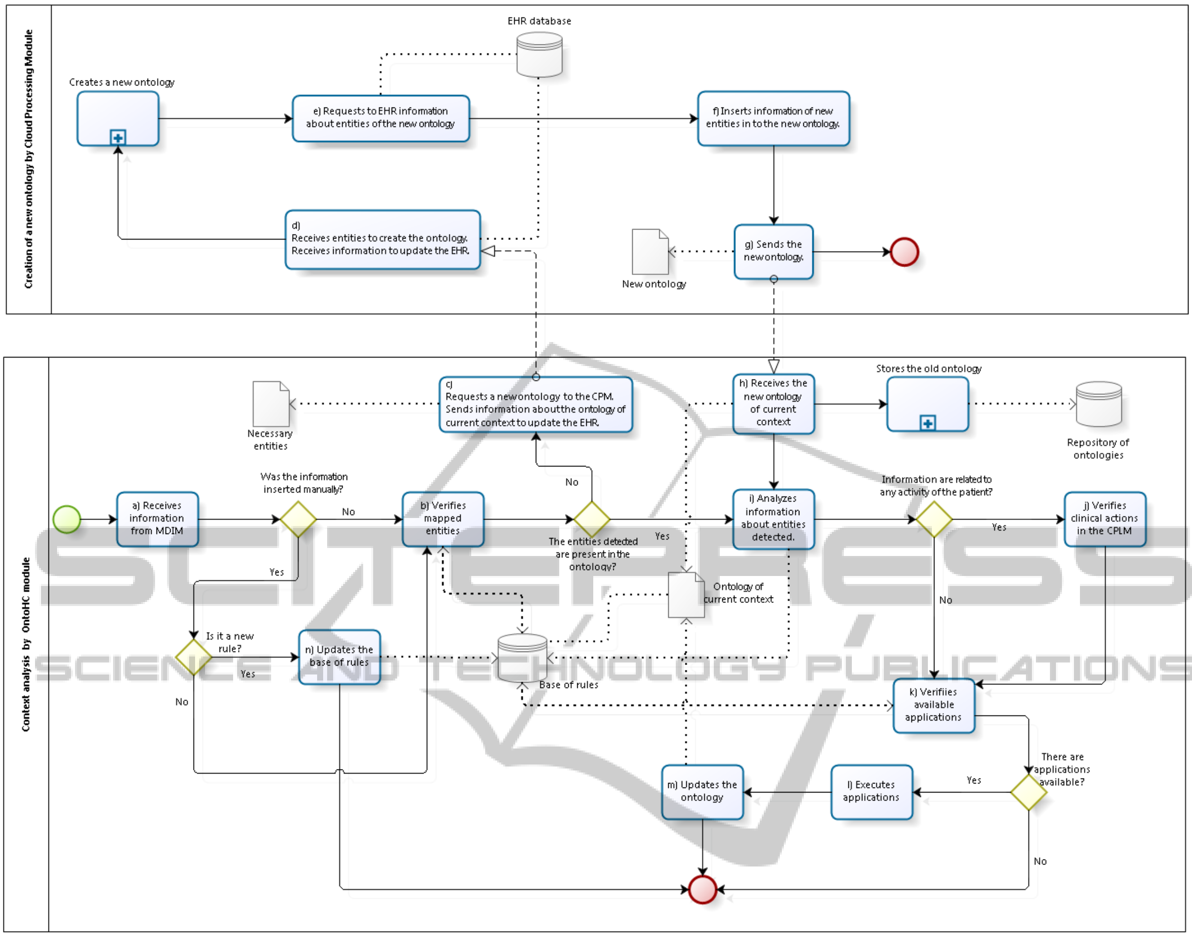

Figure 3: Process model of management of the ontology.

care plan module to search for daily activities of the patient. However, if the answer of the question of step *i* is "No", means that, there is no activity related to the patient and the flow skips the step *j* and goes to *k*.

In step *k* the system accesses the base of rules and searches for applications to be performed. This includes applications to help patients during their daily tasks (e.g.reminders of duties or cognitive games), or to assist the professionals during the treatment of the patient (e.g. reminders of medicine administration or showing of exams). If the system finds an application, the flow goes to step *l* where the application is executed, and based on how the users interact with it, the ontology is updated when they finish (step *m*).

The workflow may end in three different scenarios. First, if the information was inserted manually in the beginning of it and refers to a new inference rule or query (steps *a, n*). In this case, after update the base of rules, the flow is finished. The second scenario is when no application is found in step *k*. The last scenario is when one or more applications are executed and after that the ontology is updated by the system (steps *l, m*).

6 CASE STUDY

In this section we present a case study to validate the workflow developed in order to show how the management of the ontology is performed in the *OntoHC module*.

In the scenario, John is a patient with a minor form of Chronic Obstructive Pulmonary Disease. People who suffer from it may have their quality of life improved if they become more active. The vital signs of John should be monitored, once a sudden health crisis (e.g., drop in the level of oxygen saturation) can lead to hospitalization where the treatment will possibly be more expensive and they will have to stay a long time.

Additionally, John has a hearing impairment, moderate memory problems and need to be reminded of their tasks (e.g. taking a medicine). John's house is equipped with Tablets, PDAs, smart dispensers, vital signs checkers and sensors. These sensors are able to measure the environmental conditions (e.g. temperature and humidity).

Nancy is a professional caregiver and is responsible for creating and managing the homecare services. Among the services configured by Nancy there is an alert to remind John to take his medicine at the correct time and to add a new medicine prescribed by Jonh's physician in a dispenser.

When a prescription is registered, it is associated with an event, controlled by a timer, which is responsible for initiating the process of notification. Then, the system informs the patient in which dispenser the medicine is stored. The system must adapt its services pro-actively, at runtime, according to the preferences or needs of users. For example, John prefers to receive alerts on his mobile device without volume when he is accompanied. In case of frequent ignored reminders (e.g. due to the hearing impairment of the John), the system can send a message to Nancy to reconfigure the services to get John's attention.

When Nancy arrives at the house, the sensors detect the presence of a relevant instance of a class. Then, *monitoring and data input module* is notified and inserts information about her into an XML document and sends it to the *OntoHC module*.

After receiving the information, the *OntoHC module* uses SQWRL queries to check if the instances detected are already mapped in the ontology. In affirmative case, it checks what actions could be triggered and informs the *Notification module*. However, if the ontology does not have the classes, a new one must be created with the necessary instances, and the current should be stored in the repository, as seen in section 5.2. For this case study, we consider that the two instances (Nancy and John) are mapped in the ontology.

When the event is triggered, the *OntoHC module* searches for information about the device and dispensers of the medicine. The system creates a XML document with the necessary information for the mobile application show a reminder to John. Then, the file is forwarded to the *notification module*, responsible for sending the information about the medicine and the dispenser as a reminder to a device next to John. After display the notification, the system checks if the patient performed the task, within a range of pre-set time, defined by Nancy. This information can be captured by the sensors.

If the patient do not perform the task, the system redirects the notification to the Nancy's device and she will be able to help him. Finally, the ontology of the current context is updated, indicating that the task was performed (patient took the medicine). When the caregiver assists the patient, he can inform the completion of the task manually.

We can conclude that the system helps to reduce problems of human failures (e.g. forgetting medication). The patient is able to keep his autonomy during the treatment, reducing the intervention by the caregiver.

7 CONCLUSION

Nowadays the hospitals are facing a problem of demand of their services, where in some cases it is higher than what they can support, creating queues or a low satisfaction by the patients. One way to help solving this problem is to take the services offered by health care providers to the house of patients. Homecare environments are characterized by being a domain with all the necessary infrastructure to the patient to be treated in his own house, without the need of going to the hospital.

The development of pervasive applications helps to ensure that the services provided to patients in their houses have the same quality as those offered in hospitals. Considering this, we first presented an architecture to be used as basis to develop pervasive systems for homecare environment and also an ontology to represent the knowledge that this kind of domain.

In this paper we described the management of the ontology, showing the workflow process of the module responsible by the creation of ontologies and how the inference on it is performed. The model developed was validated through a case study. Systems developed from the architecture can help users in different scenarios ranging from small tasks like remember patients of their daily activities or cognitive games for brain exercises, until more complex applications such as to assist the professionals during their duties, suggesting actions to be taken by them.

We intend to continue this project describing the other modules of the architecture through workflow processes models and focus on some aspects like privacy and safety of information. Until this part of the project we assume that the homecare environment has the necessary infrastructure that allows the development of pervasive applications considering these aspects. Then, we intend to build a real homecare environment to develop tests and simulate real situations, where we will be able to analyse aspects like performance of the network (e.g. the time between the detection of a new instance by sensor until the creation a new ontology to be used by the system). With this, we will be able to make a better analysis of the system, and then make it available to real users.

REFERENCES

Alliance, O. M. (2001). Wireless application protocol. http://www.openmobilealliance.org/tech/affiliates/wap/wap-248-uaprof-20011020-a.pdf.

Andrade, L. M., Martins, E. C., Caetano, J. A., Soares, E., and Beserra., E. P. (2009). Atendimento humanizado nos servios de emergłncia hospitalar na percepo do acompanhante. In *Revista Eletrnica de Enfermagem*, volume 1, pages 151–157.

Antoniou, G. and van Harmelen, F. (2003). Web ontology language: Owl. In Staab, S. and Studer, R., editors, *Handbook on Ontologies in Information Systems*, pages 76–92. Springer-Verlag.

Bastiani, E., Librelotto, G. R., Freitas, L. O., Pereira, R., and Brasil, M. B. (2013). An approach for pervasive homecare environments focused on care of patients with dementia. In *International Conference on Health and Social Care Information Systems and Technologies*. Elsevier.

Chen, H. (2004). *An Intelligent Broker Architecture for Pervasive Context-Aware Systems*. PhD thesis, University of Maryland, Baltimore County.

Evchina, Y., Dvoryanchikova, A., and Lastra, J. (2012). Ontological framework of context-aware and reasoning middleware for smart homes with health and social services. In *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, pages 985–990.

Gassen, J. N., Machado, A., Thom, L. H., and Oliveira, J. P. M. (2012). Ontology support for home care process design. In *International Conference on Enterprise Information Systems*.

Grau, B. C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., and Sattler, U. (2008). Owl 2: The next step for owl. *Web Semantic*, 6(4):309–322.

Horrocks, I., Patel-Schneider, P., Boleym, H., Tabet, S., Grosof, B., and Dean, M. (2004). swrl: a semantic web rule language combining owl and rule. Available: http://www.w3.org/Submission/SWRL.

Lemos, N. D., Gazzola, J. M., and Ramos, L. R. (2006). Cuidando de pacientes com alzheimer: O impacto da doena no cuidador. In *Sociedade e Saude*.

O'Connor, M. and Das, A. (2009). Sqwrl: a query language for owl. In *OWL: Experiences and Directions*. Available: http://www.w3.org/Submission/SWRL.

Paganelli, F. and Giuli, D. (2011). An ontology-based system for context-aware and configurable services to support home-based continuous care. *Information Technology in Biomedicine, IEEE Transactions on*, 15(2):324–333.

Thirugnanam, M., Thirugnanam, T., and Mangayarkarasi, R. (2013). An ontology based system for predicting disease using swrl rules. *International Journal of Computer Science and Business Informatics*, 7(1).

Zarghami, A., Eslami, M., Sapkota, B., and van Sinderen, M. (2011). Toward dynamic service provisioning in the homecare domain. In *The 5th International Conference on Pervasive Computing Technologies for Healthcare*. Available: http://www.w3.org/Submission/SWRL.