Towards an observer/controller and ontology/rule-based approach for pervasive healthcare systems

Amina Hameurlaine, Kenza Abdelaziz, Philippe Roose, Mohamed-Khireddine Kholladi

Towards an observer/controller and ontology/rule-based approach for pervasive healthcare systems

Amina HameurLaine*

MISC Laboratory,
Department of Computer Science and its Applications,
University of Constantine 2 – Abdelhamid Mehri,
Constantine, 25000, Algeria
Email: amina.hamerelain@hotmail.fr
*Corresponding author

Kenza Abdelaziz

Modeling and Engineering,
Pierre et Marie Curie University,
Paris, 7500, France
Email: abdelaziz.kenza@gmail.com

Philippe Roose

LIUPPA/UPPA,
IUT de Bayonne et du Pays Basque,
2 Allée du Parc Montaury,
Anglet, 64600, France
Email: Philippe.Roose@iutbayonne.univ-pau.fr

Mohamed-Khireddine Kholladi

Department of Mathematics and Computer Science,
University of El Oued – Echahid Hamma Lakhdar,
El Oued, 39000, Algeria
Email: kholadi@yahoo.fr

Abstract: This paper proposes a new design approach for U-Healthcare systems. Our approach consists in combining the generic observer/controller architecture of organic computing, ontologies and rule-based paradigms. This combination brings three significant benefits. First, it allows keeping the system highly supervised and controlled by the observer/controller. Secondly, it enables the system to reason upon useful contextual information gathered from different and heterogeneous entities, and deduce new situations that require adaptation of the system's behaviour. Finally, it enables the system to adapt its behaviour dynamically to the context change by reasoning about this context and selecting the appropriate service to a specific user.

Keywords: pervasive computing; U-Healthcare systems; context-awareness; context-aware systems; context modelling; context reasoning; ontologies; rules; observer/controller architecture.

Reference to this paper should be made as follows: HameurLaine, A., Abdelaziz, K., Roose, P. and Kholladi, M-K. (2017) 'Towards an observer/controller and ontology/rule-based approach for pervasive healthcare systems', *Int. J. Ad Hoc and Ubiquitous Computing*, Vol. 26, No. 3, pp.137–156.

Biographical notes: Amina HameurLaine received her PhD in Computer Science from Constantine 2 University, Algeria. She was supervised by Professor Mohamed-Khireddine Kholladi. Her work is mainly focused on pervasive computing, dynamic adaption, context-awareness, and ontologies.

Kenza Abdelaziz got her Master degree in Artificial Intelligence from University of Pierre et Marie Currie Paris (France) in 2013. Her work is mainly focused on pervasive computing, ontologies and semantic data.

Philippe Roose received his PhD in 2001 and is now an Associate Professor at the University of Pau. He is Head of the Computer Science Department and Leader of the LIUPPA/T2I research team. He has published several international papers and holds three international patents. His research activity is mainly about middleware, context aware applications and information systems. He his co-inventor of the Kalimucho middleware presented in CES Las Vegas and World Mobile Congress in 2014. He has several (but not limited) international collaborations, mainly with Maghreb and Latin America.

Mohamed-Khireddine Kholladi graduated in Computer Science from INSA Lyon in France. He was Rector of the University Echahid Hamma El Oued Lakhdar. He was director MISC Research Laboratory of the University Abdelhamid Mehri 2 Constantine Constantine. He is still President of the Science and Development Association (ASD) at the University of Brothers Mentouri Constantine 1 Constantine. He is a member of several international networks such as: ACIT (Arab Network of ICT), CCIS (Arab Network of ICT), Cassini in France, etc. His research points are: geographical information systems (GIS), computer graphics, image processing and synthesis, knowledge bases, metaheuristics, complex systems, national archives, ICT, etc.

1 Introduction

In recent years, pervasive systems have become a very active area of research in which we are surrounded by a multitude embedded and smart devices; including sensors, smart phones, tablets, cameras, etc.; interacting and exchanging information using wireless communication technology. Pervasive computing (Weiser, 1991) makes information available anywhere and at anytime, and offers several smart systems aiming to provide us various significant services in our life (Agoston et al., 2000). Ubiquitous or pervasive healthcare systems (U-Healthcare systems) are one of the main application areas of pervasive computing that provide us several services; such as monitoring health and wellbeing of patients anytime and anywhere via smart mobile devices. With the progress of ubiquitous technologies, more smart services can be provided in a smart home environment that allows patients to live safely and independently in their homes.

For providing such services, this system has to react rapidly by analysing dynamically not only the health measurements of patient, but also by analysing his current location, his profile, his activities, and his physical environment. With the invasion of mobile devices in our life, pervasive healthcare systems can provide us mobile services or applications that allow monitoring health and wellbeing of patients anytime and anywhere. However, the obstacles related to the limited resources of mobile devices, such as battery lifetime, hinder the service continuity in this kind of systems. Therefore, device resources have to be taken into account during the execution of these systems, and the service continuity on mobile devices must be ensured. In order to deal with the different resources presented in the environments, this kind of system must adapt their behaviours according to the context change (Baldauf and Dustdar, 2007). They must be able to detect context information that comes from different and heterogeneous entities, infer new situations from this context, and provide the appropriate services to the user at the right time, in the right place and with the right manner.

Ontologies seem to be one of the most appropriate approaches to build context-aware systems and have become a very popular research topic in knowledge representation (Strang and Linnhoff-Popien, 2004). They provide effective mechanisms to represent contextual information due to their high and formal expressiveness. Ontologies are applied in several research fields as news summarisation (Lee et al., 2005), medical information systems (Orgun and Vu, 2006), and healthcare applications (Wang et al., 2010). Besides, Rule-based approach is used as a way to interpret information in a useful way. They are used to ascribe meaning to, and to reason about data (Eiter et al., 2008). Using rules with ontologies is becoming a suitable solution for representing and reasoning upon contextual information in ubiquitous healthcare scenarios; as clinical or smart-home environments; where heterogeneous sources work together.

On the other hand, the complexity of pervasive healthcare systems is steadily increasing. In these systems, there is a growing variety of mobile computing devices, which are highly connective and can be used for different tasks in dynamically changing environments. Like in other complex systems, the global behaviour of U-Healthcare systems might be unexpected. Owing to this complexity, innovative ideas for the design and the development of these systems must be found (Schmeck, 2005). Autonomic Computing and Organic Computing paradigms have been introduced in order to overcome the rising complexity of computing systems. Organic computing paradigm aspires to the development of robust, flexible and highly adaptive computing systems. Observer/controller architecture presents one of the most based concepts of organic computing. In this architecture, the system is observed and potentially controlled in order to comply with the objectives given by the user or the developer. In addition, by using observer/controller architecture, organic computing can make embedded systems more life-like by providing them with so-called self-X properties (Kluge et al., 2008).

In this context, we aim to combine the observer/ controller (O/C) architecture of organic computing paradigm with ontologies and rule-based approaches in

order to get most of their benefits for realising new design approach for pervasive healthcare systems. Our approach permits to keep systems highly supervised and controlled by the observer/controller. Such supervision allows detecting, analysing and reasoning upon useful contextual information using a hybrid reasoning engine based on the ontology and rules. Moreover, it allows adapting automatically the behaviour of system according to the current context in order to provide the appropriate service to the user.

The rest of the paper is organised as follows. Section 2 presents the basic concepts of the technologies involved in our approach, while related work is discussed in Section 3. Section 4 presents a motivation about our proposition; Section 5 describes in detail our approach. Implementation details are given in Section 6, and Section 7 presents discussion about our approach. Finally, Section 8 concludes the paper and gives future trends.

2 Basic concepts

2.1 Context modelling and context reasoning

Owing to the characteristic of reacting upon context changes, context-awareness plays a central aspect in the development of ubiquitous systems. Developing context-aware systems in pervasive computing should be supported by adequate context information modelling and reasoning techniques (Bettini et al., 2010). These techniques reduce the complexity of those systems and improve their maintainability. Indeed, context modelling is a fundamental step for context-aware system adaptation. It allows representing and storing context data in a machine processable form. A context model should be able to represent context information and support reasoning techniques. The latter are used to derive new context information from existing information and reason about high-level context abstractions that model real world situations.

Strang and Linnhoff-Popien (2004) summarised the most interesting context modelling approaches in the literature which are:

- key-value model
- mark-up scheme models
- graphical models
- object-oriented models
- logic-based models
- ontology-based models.

Key-Value represents the simplest data structure for modelling context information as a couple (attribute, value). This model is easy to implement but it is not convenient for complicated structure and does not allow a good reasoning on context. *Markup scheme modelling approaches* are based on XML and use hierarchical data structure consisting of markup tags with attributes and content. The most used examples of these models are the composite capabilities/preference profile (CC/PP) (W3C, 1999) and user agent profile (WAPFORUM, 2001). These models are simple, flexible and can be used in structured and distributed computer systems. However, it has difficulty in defining the complex relationships between elements of information. Unified modelling language (UML) (Sheng and Benatallah, 2005) and ORM (Henricksen et al., 2003) are also used to model the context owing to its generic structure. These models are simple but less formal than the Markup scheme models. *Ontology-Based Models* seems to be the most suitable tool for context modelling owing to their high and formal expressiveness that can be used to describe complex context data that cannot be represented, for example, by simple languages like the CC/PP. In addition, ontologies provide possibilities for using reasoning techniques, such as inference rules.

2.2 Ontology and rule-based approaches

Currently, the integration of ontologies and rules has received important attention in the research on ontologies and the semantic web (Rosati, 2006). Ontology (Fensel, 2004) can be defined as a formal explicit definition of a shared conceptualisation and has many benefits such as knowledge sharing, logic inference and knowledge reuse. By using ontology, it is possible to provide general expressive concept and support for syntactic and semantic interoperability. Ontology consists typically of a number of classes, a number of relations (sometimes called properties) between these classes, and a number of instances and axioms. These elements are all expressed using some logical language. Ontology-based approach is a very promising instrument for modelling contextual information; owing to their high and formal expressiveness. In addition, ontologies are used to assist in communication between humans, to achieve interoperability, and to facilitate communication among software systems (Uschold and Jasper, 1999). They are very powerful and applicable in ubiquitous environment (Strang and Linnhoff-Popien, 2004).

Rule-based approach is used as a way to interpret information in a useful way. Using this approach gives to the system the ability to determine which option should be taken in a specific situation by using a set of rules of the form *IF some condition THEN some action* (Ireson-Paine, 1996). Using rules with ontology provides to the systems an efficient mechanism for discovering and generating automatically new relationships between concepts, based on existing ones (W3C, 2013).

2.3 The generic observer/controller architecture of organic computing

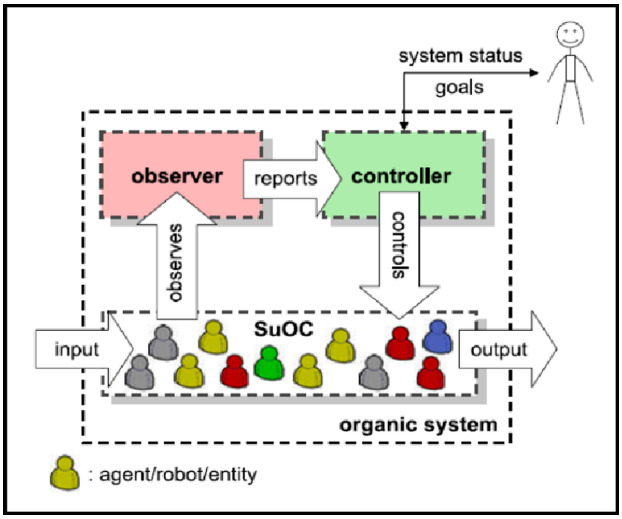
Organic computing (OC) (Schloer, 2004) is becoming the most recent vision for designing complex systems. The main objective of OC is to make systems more life-like by giving them the self-x proprieties of autonomic computing (Kephart and Chess, 2003); such as self-organisation, self-configuration, self-healing, self-optimisation and self-protection and can also expand towards self-adaptivity and

self-awareness. One of the key statements of the OC is the following: "It is not the question whether self-organised and adaptive systems will arise but how they will be designed" (Schmeck, 2005). The most significant aspect of OC systems is the autonomic adaptability to the dynamic environmental changes in response to the human needs; by giving the system ability of sensing their current situation and providing appropriate responses to dynamically changing environmental conditions (by giving them adequate degrees of freedom).

The generic observer/controller architecture (Branke et al., 2006; Tomforde et al., 2011; Richter and Christian, 2006) is the basic concept for designing organic systems. In this paradigm, the system is observed and potentially controlled in order to comply with the objectives given by the user or the developer. As shown in Figure 1, this architecture contains three major components: system under observation and control (SuOC), observer and controller. The observer/controller uses a set of sensors and actuators to measure system variables and to influence the system. The observer/controller and the SuOC form the so-called organic system. The SuOC which constitutes productive system that serves a specific purpose; is supervised by the observer and the controller. An observer/controller loop enables adequate reactions to control the emerging global behaviour resulting from interactions between local agents. As described in Richter and Christian (2006) (Branke et al., 2006), the aim of the Observer is to collect and aggregate available information about the SuOC by monitoring the system. The information coming from the SuOC constitutes raw data that is stored in a log file for every loop of observing/controlling the SuOC. This data is pre-processed in order to select the relevant data which is required to compute aggregated system-wide parameters. The pre-processes data is analysed to provide a global description of the current state of system and also used to predict the future system state. The results of pre-processing, analysing and predicting process is aggregated to situation parameters. Then, the Observer reports description of the currently observed situation to the controller. The controller evaluates this situation and takes appropriate actions whenever it is necessary to influence the underlying system to meet the system goal. Using this control loop, an organic system will over time adapt to its changing environment. It is obvious that the controller could benefit from learning capabilities to tackle these challenges.

Several works from varying application domains have adopted the generic observer/controller architecture for developing their own systems. In Tomforde et al. (2011), a survey which presents research projects that use the observer/controller paradigm to solve technical problems from various domains is presented. For instance, the organic network control (ONC) (Tomforde et al., 2009) is a system that has been proposed in order to dynamically adapt parameter configurations of data communication protocols to the change of environmental conditions. In The MeRegioMobil project (Becker et al., 2010), an hierarchically structured observer/controller architecture is applied to control a smart-home, to increase the energy efficiency, and to avoid overloads of the low voltage grid.

Figure 1 Observer/controller architecture (see online version for colours)



Source: Schmeck et al. (2011)

3 Related works

Several works have been proposed for adapting pervasive systems to the dynamic change of context. MUSIC (Romain et al., 2009) is the most well-known autonomous platform supporting self-adaptative mobile and context-aware applications. This platform can be adapted to the dynamic changes of environment (e.g., location, network connectivity) in order to satisfy the user requirements and device properties (battery, memory, CPU). Other recent works such as CAMSPF (Pan et al., 2013) proposed a cloud based framework for adapting mobile services in pervasive computing environment by constructing a private mobile service adaption agent for each mobile user in the cloud.

However, these works have not taken into account the contextual information that can be gathered by bio-sensors; namely vital signs such as blood sugar and blood pressure measurements. Therefore, they cannot be used in healthcare field. In addition, those works do not hold the services selection and then, they do not allow providing the appropriate service to the user.

Other interesting works are proposed in the field of pervasive healthcare system. They have demonstrated the amenability of ontologies and Rules for building context-aware pervasive computing systems. Catarinucci et al. (2012) have proposed a context-aware infrastructure for ubiquitous and pervasive monitoring of heterogeneous healthcare-related scenarios. Their system is based on ontology representation, multi-agent paradigm and rule-based logic. They have used an ontology and inference rules for modelling and reasoning on context information. In the work (Kim and Chung, 2014), authors proposed a U-healthcare service system and an ontology-based healthcare context information model to implement a

ubiquitous environment. In the other work (Zeshan and Mohamad, 2012), an ontological model for organising the knowledge in the heterogeneous domain of embedded devices and complex healthcare systems has been proposed. In Ko et al. (2006), authors have proposed an intelligent context-aware system based on ontology which can be reused in any system that uses bio-context in pervasive environment. In Lee and Kwon (2013), authors proposed a platform for the smart home healthcare systems based on ontologies and rule-based reasoning. They have used ontologies and rule-based approach to model and reason upon context. The majority of these works have proposed interesting ontologies that can be used in the ubiquitous healthcare filed. However, they are specified to the healthcare context-aware system. Authors have modelled context according to their point of view and they did not provided a generic context model that can be used for another developer to build another context-aware system. In addition, they have used rules for reasoning upon contextual information and deducing health situation without taking into account situations related to the mobile device's resources. They have not taken into consideration the obstacles related to the limited resources of mobile devices, such as battery life time, and then they did not propose solutions to assure the service continuity on mobile devices.

Other works have used cloud computing paradigm as solution for reliving the huge burden on user's mobile computing devices and overcoming obstacles related to the performance. In Rolim et al. (2010), authors proposed an architecture to automate the process of collecting patient's vital data via a network of sensors connected to legacy medical devices, and deliver this information to the medical centre's 'cloud' for storage, processing, and distribution. In Yuriyama and Kushida (2010) authors proposed an infrastructure called sensor-cloud infrastructure which can manage physical sensors on IT infrastructure. In Doukas and Maglogiannis (2012) and Doukas et al. (2012), authors proposed platforms based on cloud computing for managing pervasive healthcare data. All these works have proposed interesting solutions for acquiring, storing and managing sensor data using cloud computing paradigm. However, they are depending on internet network. Data will be processed in the cloud and not in local devices, so if an internet connection problem will be met, the sensors data cannot be sent to the cloud and that leads the interruption of the system. In addition, according to Akyildiz et al. (2002), the energy expenditure in data communication is much more compared with data processing. Therefore, system that uses mobile computing devices such as smart phone which connect over-time to the cloud in order to send and receive data; can reduce the power of mobile devices.

We propose in this paper, a new design approach for pervasive healthcare systems; focusing on the generic observer/controller architecture of organic computing, ontologies and rule-based approaches. Our proposal approach permits to adapt dynamically the system to the current context, and can support solutions to assure the service continuity by taking into account the limited resources of mobile devices.

In the following section, we give a motivation about proposing our approach.

4 Motivation

Pervasive healthcare systems are one of the main application areas for pervasive computing that provide us several services, such as monitoring the health and wellbeing of elderly persons anytime and anywhere via smart devices. These services allow them to live safely and independently in their own homes. For providing such services, those systems must be able to learn about their environment over time, and react rapidly even if they encounter a new situation in which behavioural adaptation is required. This means that they have to adapt their behaviour to the dynamic change of context.

In order to allow for such an adaptation process, the system needs to be supervised constantly and to have the ability to reason upon contextual information. As we presented in previous sections, organic computing is an interesting paradigm for designing complex systems. Based on the generic observer/controller architecture, organic computing allows designing and developing robust, flexible and highly adaptive computing systems. In this architecture, the system is observed and potentially controlled in order to comply with the objectives given by the user or the developer. The O/C architecture has already been successfully applied for designing several systems (Becker et al., 2010; Tomforde et al., 2009), owing to its capability of supervising systems and supporting the adaptation of these systems to the change of environmental conditions. This motivates as to design new architecture for pervasive healthcare systems using the generic observer/ controller architecture in order to support the adaptation of this kind of systems.

In the other hand, in order to provide the appropriate service to the user, the pervasive healthcare system has to react rapidly on new situation by analysing and reasoning dynamically about not only the health measurements of patient, but also by analysing his current location, his profile and his physical environment. All this information falls under the notions of context. This makes context-awareness in general and context-aware adaptation in particular, an essential requirement for pervasive computing systems, especially pervasive healthcare systems. Our architecture must be able to reason upon contextual information that comes from different and heterogeneous entities, such as sensors and smart devices. Therefore, it is necessary to represent context information and to provide a high level of conceptual abstraction in order to allow reasoning about this information. By using ontologies, it is possible to provide general expressive concept and support for syntactic and semantic interoperability. They have many benefits such as knowledge sharing, logic inference, knowledge reuse. Ontologies are very promising instrument for modelling contextual information through their high and formal expressiveness. They also provide possibilities for using reasoning techniques such as inference rules. Rule-based

approach can be used as a way to interpret information in a useful way. Using this approach gives the system the ability to determine which option should be taken in a specific situation. Moreover, the amenability of both approaches (ontologies and rules) for building context-aware pervasive computing systems is demonstrated in several works. Therefore, our architecture uses ontologies and rule-based approaches to represent and reason upon contextual information, infer the current situation, and select the appropriate services to the user according to the inferred situations.

Furthermore, for monitoring the health and wellbeing of patients and elderly persons anytime and anywhere, pervasive computing in general and mobile pervasive healthcare systems in particular; provide mobile services and mobiles applications that can be used on mobile devices. These systems must not only be able to provide the appropriate service to the user, but also to assure the continuity of these services. However, resource constraints of mobile devices, such as battery lifetime of smart phones, hinder the service continuity and can interrupt the execution of services. Therefore, as other context information, the resources of mobile devices must be taken into account. Our approach should represent and reason about this information in order to deduce situations related to the mobile devices. In addition, our architecture should support solutions that can be proposed from the developer to assure the services continuity on mobile devices, such as applying mechanisms of dynamic service redeployments, including functionalities as migration, replacement, etc. (Da et al., 2014).

5 Proposition

In our works, we propose new design approach for pervasive healthcare systems; that supports the dynamic adaptation of system according to the current context. This approach aims to combine the generic observer/ controller (O/C) architecture of organic computing paradigm with ontologies and rule-based approaches in order to get most of their benefits for achieving our objectives as following:

- The complexity of pervasive healthcare systems is steadily increasing; where a multitude embedded and smart devices are highly connective and used for different tasks. This complexity calls for new innovative design ideas of these systems. As stated in Schmeck (2005); “It is not the question whether self-organised and adaptive systems will arise but how they will be designed”. The combination of O/C architecture with ontologies and rule-based-approaches provides new design approach for designing and developing pervasive healthcare systems.
- This kind of systems must be supervised over time in order to detect new situations that under which behaviour adaptation is required. The O/C architecture assures this supervision since the system will be observed and potentially controlled.
- To provide the appropriate services to the user, pervasive healthcare systems have to adapt their behaviour to the dynamic change of context. It means that those systems need to have the ability to represent, analyse and reason upon contextual information gathered from different and heterogeneous entities. Using ontologies and rule-based approaches allows giving this ability to our O/C architecture.
- Assuring the services continuity is becoming necessary in pervasive healthcare systems that use mobile devices such as smart-phones. Ontologies and rule-based approaches allow representing and reasoning on contextual information related to the mobile devices and then deducing situations of the latter.
- The O/C architecture can support the ability to take appropriate actions whenever it is necessary to influence the underling systems to meet the objectives given by the user or the developer. Therefore, the O/C can supports the dynamic service redeployments actions that allow adapting the system to the user’s devices in order to ensure the service continuity.

Before describing our approach, let we take a simple scenario.

5.1 Scenario

Mr. Adem is an elderly diabetic person who lives alone at home and needs to monitor his health. He uses an intelligent environment which includes smart devices, sensors and services. This intelligent environment constructs an Ambient System that allows him to live safely and independently within his own home.

Let’s take some examples of the system’s services.

- Mr. Adem needs to check his Blood Sugar level. He uses a bio-sensor which connects to his smart phone via Bluetooth technology. According to his blood sugar level, the system will execute the appropriate service. If his blood sugar level is out of the normal range determined by his doctor, there will be three possible cases:
  - *High blood sugar level*: In this case the health situation of person is not in danger, but he has a high blood sugar and he must take an insulin injection. The system will execute the ‘adjusting insulin dose’ service which adjust automatically the insulin dose according to the blood sugar measurement, and execute the ‘diabetes guide’ service which contains a guide of what must be done for keeping the blood sugar level in normal, such as exercises and diet.
  - *Danger blood sugar level*: In this case, his health situation is in danger, he must be transferred to the nearest emergency centre or hospital. The ‘emergency’ service must be executed in order to contact the hospital and send a brief report

contains the health situation of the person and his personal information, and then sends un SMS to his family member in order to inform them about the situation of Mr. Adem.

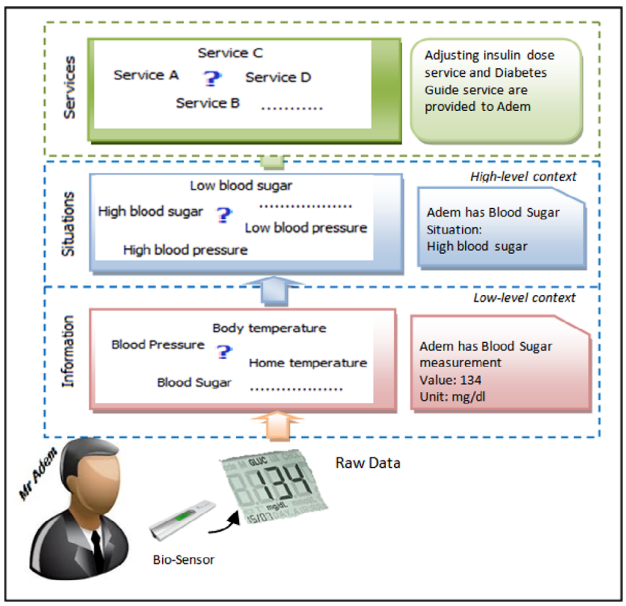
- *Low blood sugar level*: In this case, the system will execute the 'diabetes guide' which contains a guide of what he must do for adjusting his blood sugar level.

- Mr. Adem wants to take a shower. When he enters in his bathroom and activates the faucet; 'adjusting water temperature' service will be executed automatically in order to adjust the water temperature.
- In his smart home environment, when the system detects abnormal situations representing the too high temperature and the too low humidity, the 'fire station call' service is automatically triggered and the service notifies the emergency status to the nearby fire station.

As described in Adem's scenario, pervasive healthcare system provides the appropriate service to Adem according to his blood sugar situation (high, low, etc.). Therefore, this system has to reason upon Adem's blood sugar measurement and other information, such as his diabetes type, in order to determine his health situation. Indeed, as shown in Figure 2, the blood sugar measurement is gathered from a bio-sensor, it is row data and can be meaningless for system. So, how the system that processes these data could make a decision that, first determines the Adem's health situation, and second selects the appropriate service according to this situation?

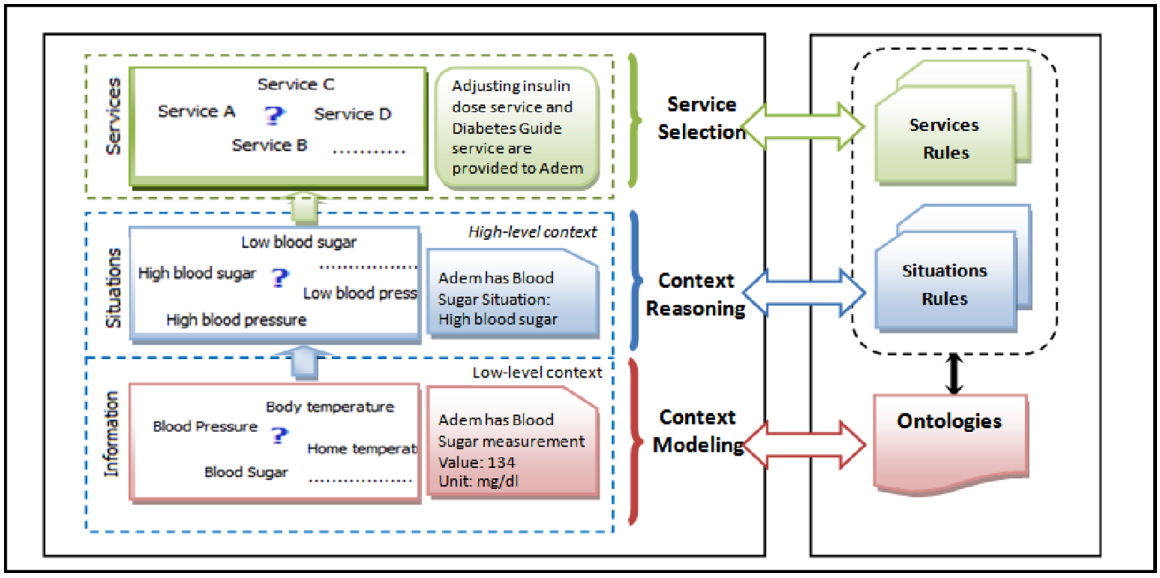
In fact, for providing us appropriate services, pervasive healthcare system has to react rapidly by analysing and reasoning dynamically about not only the health measurements of patient, but also by analysing his profile, his current location, his activities and his physical environment. In computing, this kind of concern is known as 'context-awareness'. Owing to the characteristic of reacting upon context changes, context-awareness plays a central aspect in the development of ubiquitous healthcare systems. Developing context-aware systems in pervasive computing should be supported by adequate context information modelling and reasoning techniques (Bettini et al., 2010). Data gathered from different resources, such as physical sensors, are row and meaningless and must be represented as a context information. This information is called low-level context. In this step; context modelling techniques, such as ontologies; must be used for representing context information. For example, as illustrated in Figure 3, the measurement '134' is one of row data that can be gathered from bio-sensors. This data must be represented by an adequate model in order to increase its semantic (e.g., Adem has blood sugar which its value is *134* and its unit is mg/dl). This data can be easily represented using ontologies. This representation of context will be used to infer situations that are related to the user (e.g., Adem has high blood sugar *situation*). These situations will be used to select the appropriate service that must be provided to the user (e.g., adjusting insulin dose service and diabetes guide service are provided to Adem). Inferring situations means making a high-level semantic context that requires using context reasoning techniques, like rule-based approach. Selecting the appropriate service to the user required also using these reasoning techniques; since services must be selected according to inferred situations.

Figure 2 From row data to service selection in U-healthcare system (see online version for colours)



In fact, context information can be characterised as static or dynamic (Henricksen et al., 2002). Static context information describes the invariant information that can be obtained directly from users, such as personal information. Dynamic context information describes the variant information that can be captured from different sensors, such as blood pressure sensors and temperature sensors, and also the mobile resources such as battery level. Pervasive systems focus generally on dynamic context because they need to adapt their behaviours dynamically according to the change of this context. But that does not prevent to take static context into account, such as in healthcare systems; medical information like 'disease. Owing to the diversity of context sources, context information has to be represented by an adequate uniform and extensible model that enable the system's interoperability. This model must be able to represent different categories of context information, and must support reasoning techniques that allow reasoning upon this information in order to deduce current situations relative to the user, and selecting automatically the appropriate services according to these situations. Moreover, this context model must be scalable and flexible in order to allow, on the one hand, supporting new contextual information that can be added to the environment, and on the other hand to provide an efficient generic model and structure to the developer that facilitates modelling context, designing and building context-aware systems.

Figure 3 Ontology and rule-based context model (see online version for colours)



Therefore, we propose a generic and flexible context model that facilitates modelling context and developing context-aware system. This model focuses on the hybridisation of ontologies and Rule-based approaches. It permits to represent and reason upon useful contextual information, deduce situations from this information; and allows selecting the appropriate service according to the inferred situations (see Figure 3).

5.2 *Generic ontology and rule-based context model*

5.2.1 *Context modelling*

For modelling context information, we proposed a generic ontology that can be used to represent useful context (static and dynamic context) in pervasive context aware systems. In fact, ontologies are used to capture knowledge about some domain of interest. Our ontology describes the concepts in pervasive systems domain and also the relationships that hold between those concepts. In our ontology, concepts are divided into *generic concepts* and *specific concepts*. Generic concepts can be used in any context-aware system, while specific concepts are used for a specific domain, as pervasive healthcare systems domain in our works. Figure 4 describes the structure of our ontology.

Our ontology consists of a set of classes, properties, and instances. Classes are a concrete representation of generic and specific concepts. Instances are elements of classes. Properties can be used to state relationships between individuals (object properties), or between individuals and data values (data type properties).

As shown in Figure 4, in order to facilitate the conception and the development of our ontology, we divide it into three hierarchical levels:

- *contextual information level*
- *contextual situations level*
- *contextual services level.*

These levels contain three main classes which are: *Context* class, *Situation* class and *Service* class. These classes represent generic concepts which can be used in any pervasive context-aware system that aims to provide appropriate services to the user according to the current situations.

- ***The first level*** called '*contextual information*', represents the useful context information needed to determine the appropriate service to the user. The main class of this level is '***Context***' class. This class has four sub classes: *person* class, *sensor-data* class, *location* class and *host* class.

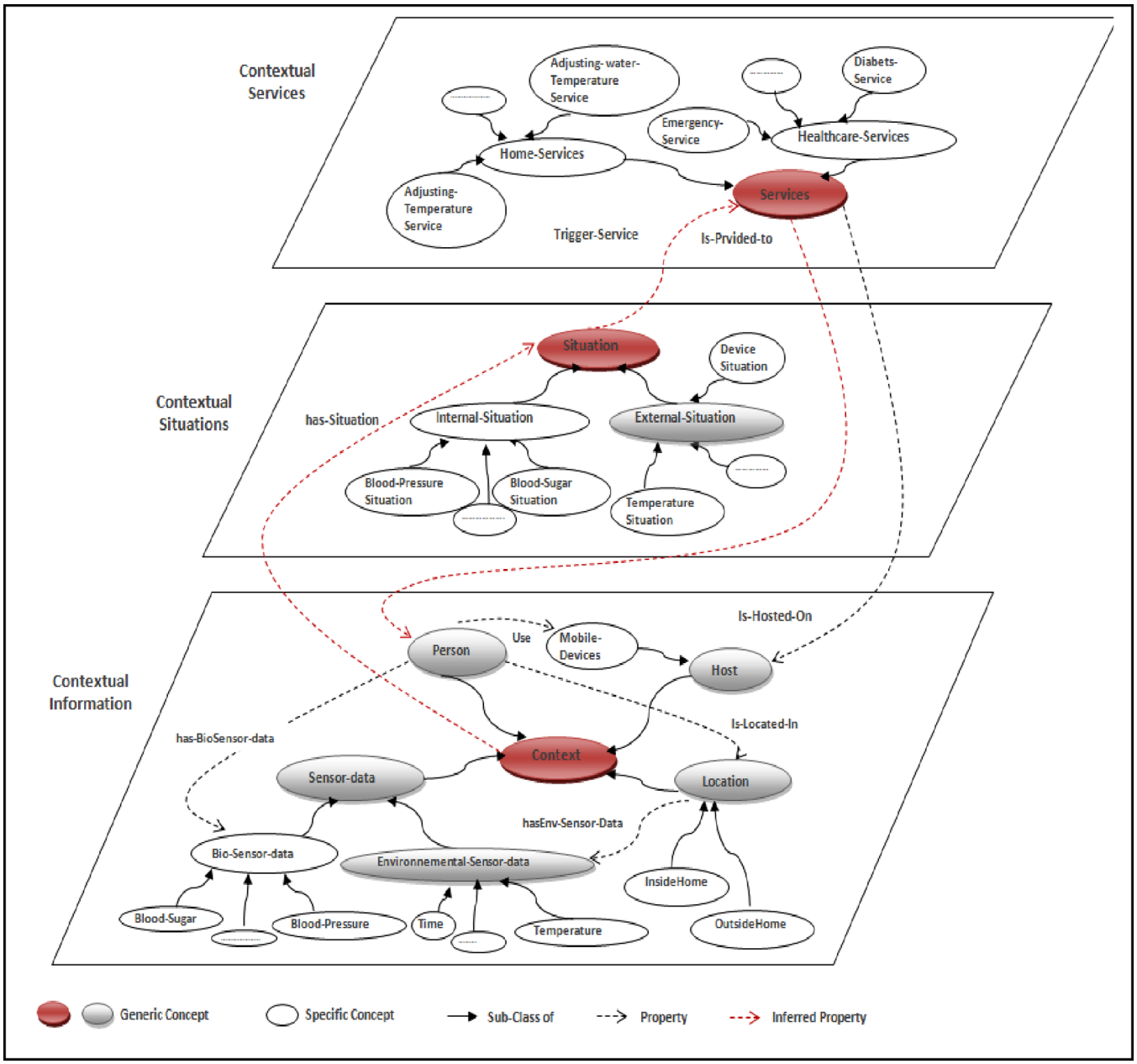
The *person class* represents user's personal information that can be specified manually, such as his profile and his medical information.

The *sensor data class* represents data which are gathered from different sensors. We have two types of sensor data:

- *bio-sensor data* represent data which are captured by bio-sensors, such as blood pressure, blood sugar and body temperature
- *environmental sensor data* represents data which are captured by environmental sensors, such as home temperature, humidity, etc.

The *location class* represents different locations where persons can be located in. This class has two sub-classes; *inside home class* represents information about person's home and its different pieces that contain environmental sensors. *Outside home class* represents other person's locations that are outside of his home and that can be considered for providing him specific services.

Figure 4 Ontology structure (see online version for colours)



The *host class* represents different hosts of services that can be proposed by the developer. For instance, services can be hosted on local devices such as mobile devices. *Mobile device class* contains all information related to the user's mobile devices such as their available resources (battery, memory and CPU).

- ***The second level*** called '*contextual situations*', represents the possible situations that we can define for each context. For instance, the home temperature context can have three situations: normal, cold, hot and very high situations. We call these situations *contextual situations* since they are depending on contextual information.

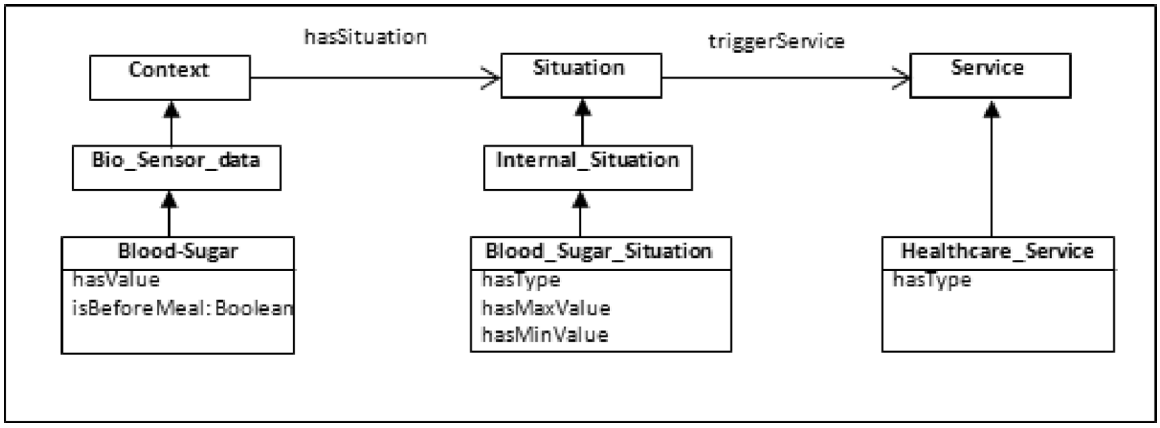
The main class of this level is '***situation class***'. This class contains tow sub classes: *external-situation class*, and *internal-situation class*. The first one represents situations that are related to the user's environment and user's devices, such as home temperature situations and battery situations. *internal-situation class* represents situations that are related to a specific domain, as the person's health state such as blood sugar situations and blood pressure situations. Each situation has data type properties (see Figure 5), as situation type, max value and a min value, that are defined by the developer and by the domain expert, like in our work; the physician. For example, *high blood sugar type1* situation is a blood sugar situation that can have 'high blood sugar type1' as a type, 126 as min value and 400 as max value.

- ***The third level*** called '*contextual services*', contains '***Service* class**'. It represents different services that can be provided by the pervasive system. Taking for example, the pervasive healthcare system that provides not only healthcare services, but also provides services

that allow patients to live safely and independently in their homes, such as controlling home temperature and water temperature. Therefore, we have created two sub classes of *service* class; *healthcare service class* presents healthcare services such as diabetes services, drug service, emergency health service, etc, and *home service class* presents services that are related to the smart home such as adjusting water temperature service, fire station call service, etc.

After modelling contextual information using ontology, we need to reason upon this information in order to deduces situations that trigger appropriate services to be provided to the user.

Figure 5 Part of ontology structure



5.2.2 Context reasoning

The main objective of proposing our ontology is not only to represent useful contextual information, but also to allow reasoning on this information using a set of rules of the form 'if–then'. These rules allow generating automatically new relationships between concepts, based on existing ones. Using this kind of rules allow the system to reason upon contextual information and infer the current situations that are related to the user. Inferring situations means generating automatically the '*has situation*' property between contextual information represented in the first level and situations represented in the second level of our ontology (see Figure 4). We have classified our rules in two categories;

- *situation rules*
- *service rules.*

Situation rules are used to infer situations from contextual information. As we discussed previously, we have two types of situations: *internal situations* and *external situations*. In the same way, we classify situation rules in two categories: *generic rules* and *specific rules*. The first ones are used for deducing *external situations*. These rules can be used in any pervasive system that includes environmental sensors and mobile devices. The *specific rules* are used to deduce *internal situations*. These rules are specified by a domain expert like a physician.

Situation rules focus on the comparison of contextual information values with the max and the min values of situations. Taking the example of *specific rules* presented in Table 1, *blood sugar type1 situations rule* means that the blood sugar measurement will be compared with the max and min values of blood sugar situations in order to infer the situation of the blood sugar context, by taking into account diabetes type, and if the blood sugar measurement is taken before or after meal.

Some of these situations such as *blood sugar situations* can trigger appropriate services that must be provided to the user. However, other situations like *device situations* can be used for proposing solutions to assure the service continuity.

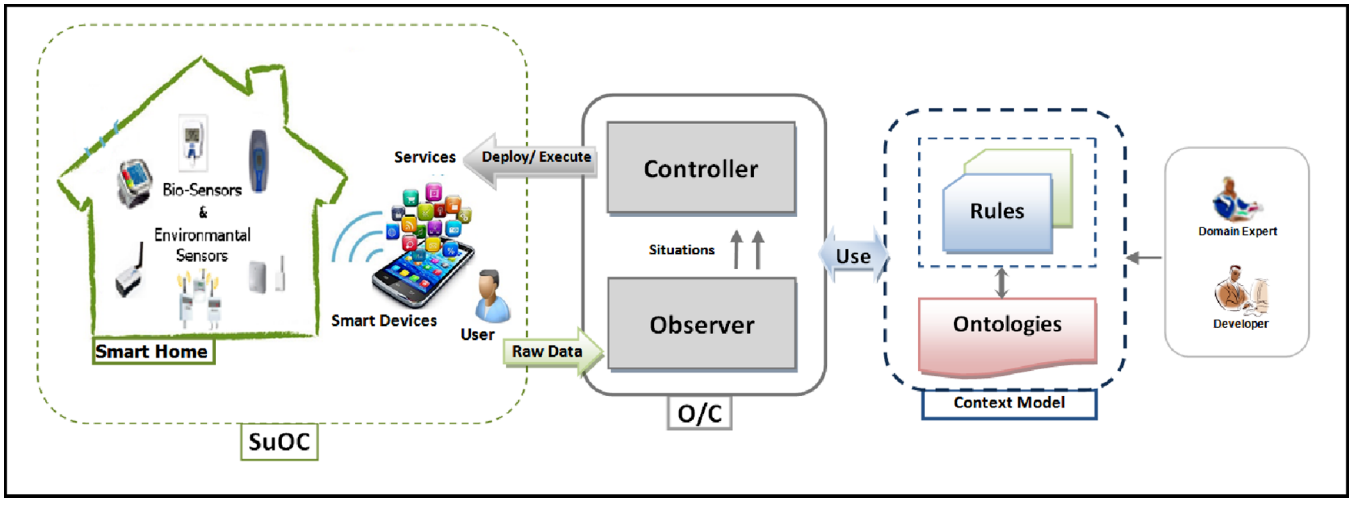
5.2.3 Service selection

According to the inferred situations, appropriate services will be provided to the user. The developer must define the set of services that can be triggered for each situation, by giving to the services the same type as situations type that can trigger these services. Then, the developer specifies a set of rules that allow selecting automatically the appropriate service according to the current situations. These rules are called '*service rules*' since they are related to the system's services. Inferring services that will be triggered by these situations, and provided to a specific user means generating automatically the '*trigger service*' property between situations and services, and '*is provided to*' property between service and person. For instance, health emergency services will be triggered by situations that have 'danger' type; including very high or very low blood sugar situations, and also very high or very low blood pressure, etc. For that, the developer must define the type of health emergency services as the same type of these situations ('danger'). The example of service rule presented in Table 1 means that the type of healthcare services are compared with the type of the deduced situations.

Table 1 Example of rules

Situation Rule	*- Rule BloodSugar_type1_Situations:* *IF* *"situation" IS A Blood Sugar Situation* *AND "situation" HAS MinValue "min"* *AND "situation" HAS MaxValue "max"* *AND IF* *"person" IS A Person* *AND "person" HAS BioSensorData "bloodsugar"* *WHERE "bloodsugar" IS A Blood Sugar* *AND "bloodsugar" HAS VALUE "true" FOR isBeforeMeal* *AND "bloodsugar" HAS Value "x" WHERE "x" IS GREATER THAN OR EQUAL TO "min"* *AND "x" IS LESS THAN OR EQUAL TO "max"* *AND "person" HAS DiabetsType1 "true"* *THEN* *"bloodsugar" HAS Situation "situation"* *"person" HAS Situation "situation"*
Service Rules	*- Rule Healthcare_service:* *IF* *"service" IS A Healthcare Service* *AND "service" HAS ServiceType "servicetype"* *AND IF* *"situation" IS AN Internal Situation* *AND "situation" HAS SituationType "situationtype" WHERE "situationtype" IS EQUAL TO "servicetype"* *AND IF* *"person" IS A Person* *AND "person" HAS Situation "situation"* *THEN* *"situation" triggerService "service"* *WHERE "service" isProvidedTo "person"*

Figure 6 Observer/controller and ontology/rule-based architecture for pervasive healthcare systems (see online version for colours)



5.3 Observer/controller and ontology-based architecture

The generic observer/controller architecture inspires us to design our architecture using the basic concepts of this architecture which are: the Observer, the controller and the SuOC. In the O/C architecture, the SuOC; which constitutes productive system that serves a specific purpose; is supervised by an observer and a controller. The Observer collects row data from the SuOC and aggregates them to situation parameters, and reports description of the currently observed situation to the controller. The controller evaluates this situation and takes appropriate actions whenever it is necessary to influence the underlying system to meet the system goal. Since our work focus on the context-awareness, we use ontologies and rule-based approaches to represent and reason upon contextual information.

Using the three main concepts of O/C architecture (SuOC, observer and controller) with ontology and rule-based approaches allow us to design new architecture that permits to collect and pre-process context, to reason on this information and then to adapt the behaviour of system according to this context.

As shown in Figure 6, our architecture is composed of four main components: SuOC, Observer, Controller, ontology and rule-based context model.

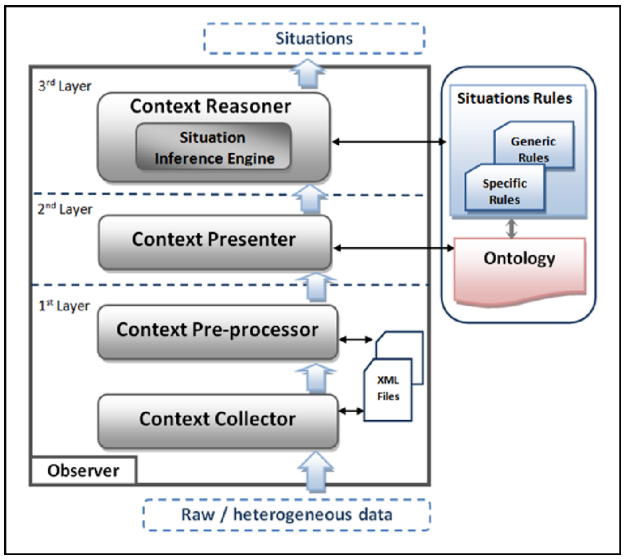
- *System under observation and control (SuOC)*: in our work, the SuOC represents user, sensors, smart devices, and services. It is responsible for providing contextual information. The SuOC is supervised by the observer and the controller components.
- *Observer*: the observer collects useful contextual information from the SuOC which represent row and heterogeneous data, analyses and interprets this data into high level information which will present situations that require adaptation of system behaviour and will be reported to the controller.
- *Controller*: based on the received situations, the Controller adapts the behaviour of system according to these situations by selecting the appropriate service that must be provided to the user and then deploys and executes this service.
- *Ontology and rule-based context model* are used by the observer/controller for representing and reasoning upon useful contextual information, and selecting the appropriate service to the user.

In the following part, we give more details about the functionalities of our architecture.

5.3.1 The observer

To provide the appropriate service to the user, the observer must be able to collect useful contextual information and to reason upon this information. The observer collects row data from the SuOC, represents these data using ontologies for giving more semantic, and then deduce the current situations using the *Situations Rules*. As shown in Figure 7, the observer has three layers where each layer generates data that will be used in the next layer. The first one presents raw data that are gathered from the *SuOC*. The second one presents semantic information represented on the ontology, and the third one presents the inferred information (situations).

Figure 7 The observer architecture (see online version for colours)



5.3.1.1 First layer

The first layer has two objectives; the first one is to collect context data, and the second one is to pre-process these data. In fact, information from physical sensors, called low-level context and acquired without any further interpretation, can be meaningless, trivial, vulnerable to small changes or uncertain (Ye et al., 2007). Therefore, this information should be pre-processed and represented by an adequate uniform context model.

In this layer, we have two modules; context collector and context pre-processor.

- *Context collector*

As in the O/C architecture (Branke et al., 2006), the ***context collector*** has the same role of the *monitor*. It is responsible for collecting static and dynamic context (personal information, device's information and sensor data) provided by the SuOC, via different interfaces like; *sensor interface* that gets data from sensors, *profile interface* that gets personal information…etc. If personal information is stored in an external database, the collector uses SQL query for collecting this information. Like in Kim and Chung (2014) and Kim et al. (2014), the collected information will be integrated to make low-level context and stored as format that can be convertible into OWL file after parsing process, such as XML forma.

Information gathered from physical sensors, can be meaningless, trivial, vulnerable to small changes, or duplicated. This means that theses data must be pre-processed before they are used in the reasoning phase.

- *Context pre-processor*

The *context pre-processor* is responsible for analysing sensor data in order to remove duplicated information and unify sensor data that can have different measurements units. Take for example; blood sugar measurements gathered from bio-sensors. The measurement unit used by these sensors can either have a weight dimension (mg/dl) or a molarity (mmol/l) (Conversion of Glucose Values from mg/dl to mmol/l, 2009). The international standard way of measuring blood glucose levels are in terms of a molar concentration, measured in mmol/l. However, in USA, blood sugar level is measured in mg/dl. In our work, measurement units must be convenient with units used in situations (min and max values). Therefore, the *context pre-processor* converts these data using standard conversion methods, such as blood sugar conversion method used in (Conversion of Glucose Values from mg/dl to mmol/l, 2009):

$$\text{mg/dl} \times 0.0555 = \text{mmol/l}$$

$$\text{mmol/l} \times 18.0182 = \text{mg/dl}.$$

5.3.1.2 Second layer

The main objective of this Layer is to represent context information using ontology-based model. In fact, information which is generated from the first layer, called low-level context is heterogeneous and can be meaningless. Using this context directly from the system makes an obstacle to reason about this information. One solution to avoid such problem is to increase the semantic of low-level context. By using ontologies, it is possible to provide general expressive concepts and supports for syntactic and semantic interoperability. For that, the ontology that we described previously is used in this layer in order to represent this information as a context. The context information must be consistent with our ontology, that's why the ***context presenter*** converts this context into a triple pattern in OWL format using a parsing process; and then inputs it as data into the ontology. Once the context is represented in ontology, the ***context reasoner*** is triggered.

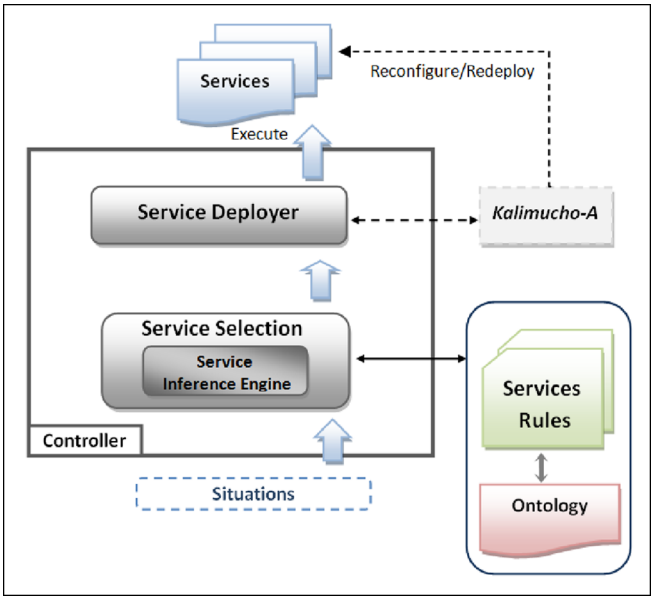
5.3.1.3 Third layer

In this layer, the ***context reasoner*** is responsible for reasoning upon context and deducing situation that will be reported to the controller. In general, context-aware systems based on ontology modelling gets low-level context from context provider or context resources in ubiquitous environment, reasons upon this context, and derives it to high-level context using an inference engine (Cabitza et al., 2005; Chen et al., 2003; Kim and Chung, 2014; Kim et al., 2014; Ko et al., 2006). Therefore, the *context reasoner* has to use an inference engine for being able to reason upon context information. We call this inference engine as situation inference engine. By using this latter, the *context reasoner* executes the *situation rules* and infers the current external/internal situations from information represented in the ontology according to the *situation rules*. These situations are related to the contextual information generated in the first layer, any change in this information could change the deduced situation in this layer. After inferring these situations, the context reasoner transfers them to the controller.

5.3.2 The controller

As show in Figure 8, the Controller is composed of two modules; the ***service inference engine*** and the ***service deployer***. The *service inference engine* is triggered by the *situations* that are reported by the observer. It determines which service to run according to the current situations by applying the *service rules*. The list of services which will be provided to the user and the situations of his mobile device resources (e.g., battery situation) are transferred to the ***service deployer***. The latter is responsible for executing services and assuring the continuity of these services on mobile devices.

Figure 8 The controller architecture (see online version for colours)



In fact, obstacles related to the limited resources of mobile devices, as battery lifetime; hinder service continuity in pervasive systems that use mobile devices for deploying services. Therefore, the *service deployer* has to take into account the limited resources of mobile devices and support solutions that assure the continuity of services. For instance, in case of full battery situation of user's mobile device, the *service deployer* will execute services; which are selected by the *service inference engine*, on the user's mobile device. While in case of low battery situation, it must hunt for solutions that allow assuring the service continuity. One solution to guarantee that is applying mechanisms of dynamic service reconfiguration and redeployments, including functionalities as migration, replacement, etc. (Da et al., 2014). Those mechanisms allow adapting services and mobile applications to the limited mobile

resources; by reconfiguring the structure of applications, or by migrating services from a device to other devices. The *service deployer* can use those mechanisms for assuring the service continuity. Actually, we have not developed our mechanisms that allow reconfiguring and redeploying services, but we propose to use existing works that are focusing on these mechanisms. In Cassagnes et al. (2009), Da et al. (2014) and Keling (2014), authors proposed software architecture for limited mobile devices, called *Kalimucho-A*. This architecture allows adapting applications in runtime to the dynamic change of device resources by reconfiguring applications and redeploying services/components of this application or moving them on other devices. In this architecture, applications are composed of a set of services, and each service is composed of a set of components.

Kalimucho-A allows not only the dynamic reconfiguration and redeployment of applications, but also to take the appropriate adaptation action, even it reconfigures the structure of application by changing, adding, or removing their components or, migrates their components from devices to other devices. However, this architecture does not hold the service selection, and then, it does not allow providing the appropriate service to the user. Therefore, we propose to use this architecture in our work in order to benefit from its adaptation mechanisms to assure the service continuity regardless the limited resource of mobile devices. As shown in Figure 8, the *service deployer* will use this software architecture, in case of low device resources situations, in order to assure the service continuity. *Kalimucho-A* will be triggered by the *service deployer* when this latter receives *low resources situations* from the *observer*. *Kalimucho-A* retrieves information related to the services that must be adapted and their hosts. Then, it will hunt for a solution to redeploy or reconfigure theses services. The dynamic services redeployment and reconfiguration procedure is out the scope of this paper, since it is detailed in Louberry (2010) and Louberry et al. (2011).

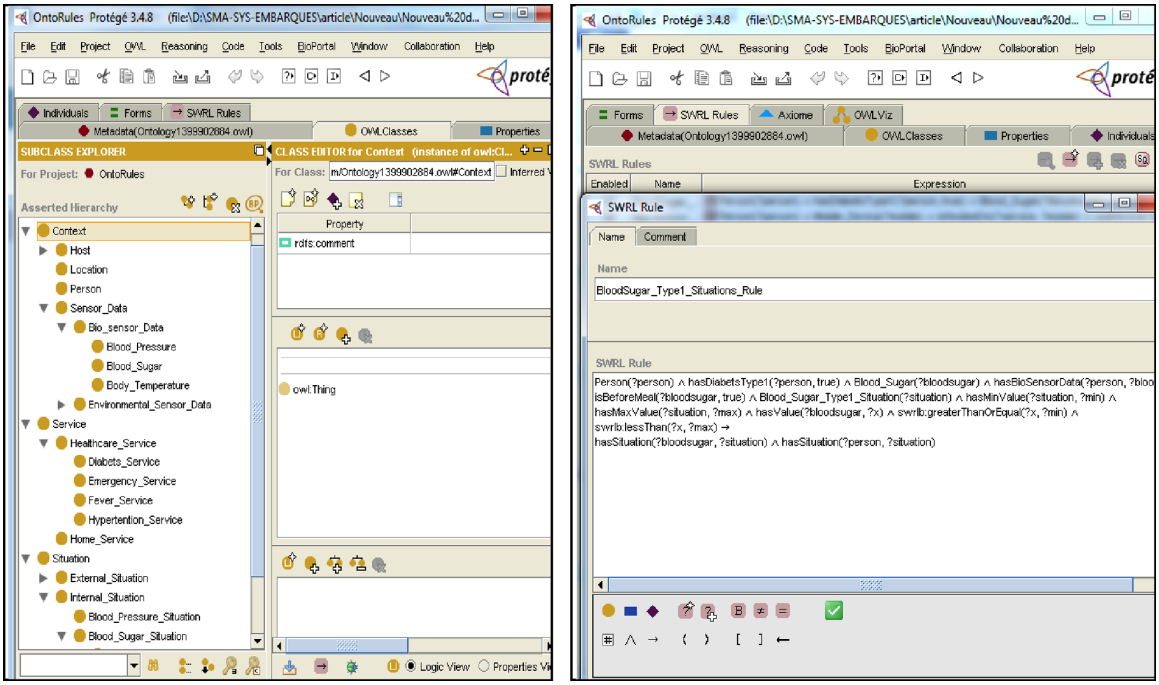
To verify the usefulness of the proposed architecture, we implemented a prototype of this architecture. Through the demonstration of the developed prototype, we know that using ontology and rule-based context model with the observer/controller can provide the current situation and the appropriate services to the user.

6 Prototype implementation

This section describes implementation details of our prototype.

Firstly, based on OWL and SWRL languages, we have implemented our Ontology and rule-based context model using Protégé tool (see Figure 9). Protégé (Horridge et al., n.d.) is an open-source platform that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontologies. In addition, it implements a rich set of knowledge-modelling structures and actions that support the creation, visualisation, and manipulation of OWL ontologies and SWRL Rules. OWL (Smith et al., 2004) is a general purpose ontology language that contains all the necessary constructors to formally describe most of the information management definitions: classes and properties, with hierarchies, and also range and domain restrictions. The semantic web rule language (SWRL) (Horrocks et al., 2004) extends the set of OWL axioms in order to include conditional rules (Horn clauses), of the form *if... then* ...SWRL allows users to write rules that can be expressed in terms of OWL concepts to provide more powerful deductive reasoning capabilities than OWL alone.

Figure 9 Protégé views (see online version for colours)



After having implemented our context model, we have used Java environment to implement the prototype of the observer/controller architecture that can use our context model to represent and reason upon contextual information, and to select the appropriate service to the user.

In this section, we aim at demonstrating that our proposal observer/controller and ontology-rule-based architecture has the ability to infer situations and determine services that must be provided to the user according to his current situations. To verify the usefulness of the proposed architecture, we have taken a set of examples of contextual information, situations and services. Tables 2–4 present some of these examples that serve as a good illustration of the effectiveness of our approach. These examples concern different diabetic persons of type 1, who use bio-sensors to measure their blood sugar level. Theirs measurements will be treated from our approach to provide them the appropriate services according to the blood sugar level. The services will be executed on their mobile devices. Therefore, the battery level of mobile devices will be treated in order to deploy and insure the services continuity on these devices. Table 2 illustrates some of blood sugar measurements and battery level examples.

We have also given a set of examples of blood sugar situations that can be specified by a physician, and battery situations that can be provided by the developer (see Table 3). Each situation has a *situation type*, *min value* and *max value*. The situation type is used to specify for each situation the type of services that can be triggered by this situation. This service must have the same type given to that situation. *Min value* and *max value* present the range of each situation that is specified by the physician and the developer. Examples of healthcare services are cited in Table 4.

Table 2 Examples of contextual information

Person	*Blood sugar measurement* (ml/g)	*Mobile device's battery level* (%)
Adem	250	15
Ali	68	52
Alice	50	90
Anis	440	60
Iness	20	65
Karim	120	25
Lilya	80	27
Lina	480	68
Linda	55	35
…	…	…

Table 3 Examples of situations

		Situation type	*Min value* (ml/g)	*Max value* (ml/g)
Internal situations: Blood sugar type 1 situations	Normal blood sugar	Normal blood sugar type 1	90	126
	High blood sugar	High blood sugar type1	126	400
	Low blood sugar	Low blood sugar type 1	51	90
	Very high blood sugar	Danger	400	800
	Very low blood sugar	Danger	0	50
			(%)	(%)
External situations: Battery situations	Low battery	…	0	30
	Full battery	…	30	100

Table 4 Examples of services

Healthcare services		*Service type*
Diabetes services	Adjusting insulin dose	High blood sugar type1
	Guide 1	High blood sugar type1
	Guide 2	Low blood sugar type 1
Emergency service	Emergency call	Danger
	Family call	Danger

To evaluate the usefulness of our approach to reason upon the examples represented in Table 2–4 we have used three interfaces to simulate these examples (see Figure 10):

- *contextual information* interface is used to simulate the contextual information
- *situation* interface is used to simulate the situation examples
- *service* interface is used to simulate the service examples.

As proposed in our architecture, the principal role of the ***observer*** is to supervise the SuOC, which represents users, sensors, smart devices, and services. The ***observer*** collects and pre-processes context information from the *SuCO* and stores it in an XML file. Then, it represents and reasons upon this context using the Ontology and rule-based context model in order to deduce the current situations that are reported to the ***controller***. The latter is responsible for providing the appropriate service to the user according to the inferred situations. In fact, the ***observer*** is composed of four main modules: *context collector*, *context pre-processor*, *context presenter* and *context reasoner*. Our prototype must demonstrate the performance of those modules. Therefore, once we click on the *test button* of the contextual information interface, the procedure of the observer and Controller is starting automatically as shown in Figure 10.

(1) Contextual information is collected from the SuCO by the *context collector*, and it is stored in an XML file. This means that the information simulated via the prototype interfaces is stored automatically into an XML file. (2) This information is pre-processed by the *context pre-processor*, and (3) converted into a triple pattern in OWL format and inserted in our ontology by the *context presenter* using a parsing process. Then, (4) the *context reasoner* uses the *situation inference engine*; which is based on Pellet inference engine (Clark et al., 2007), executes the *Situation Rules* presented in Section 2.2, and infers the current situations. (5) These situations are transmitted to the *controller*. The latter uses the *service inference engine*, which is based on Pellet inference engine too, and applies the *service rules* in order to determine the appropriate services according to the deduced situations. Table 5 presents some of situations and services that are inferred for each person. (6) The ***controller*** is responsible for deploying and insuring the service continuity on mobile devices. Therefore, it uses the service redeployment mechanisms proposed in the ***Kalimucho-A*** framework (Da et al., 2014; Keling, 2014). The demonstration of the service deployment is out of the scope of this paper.

Figure 10 Observer/controller and ontology/rule based architecture simulation (see online version for colours)

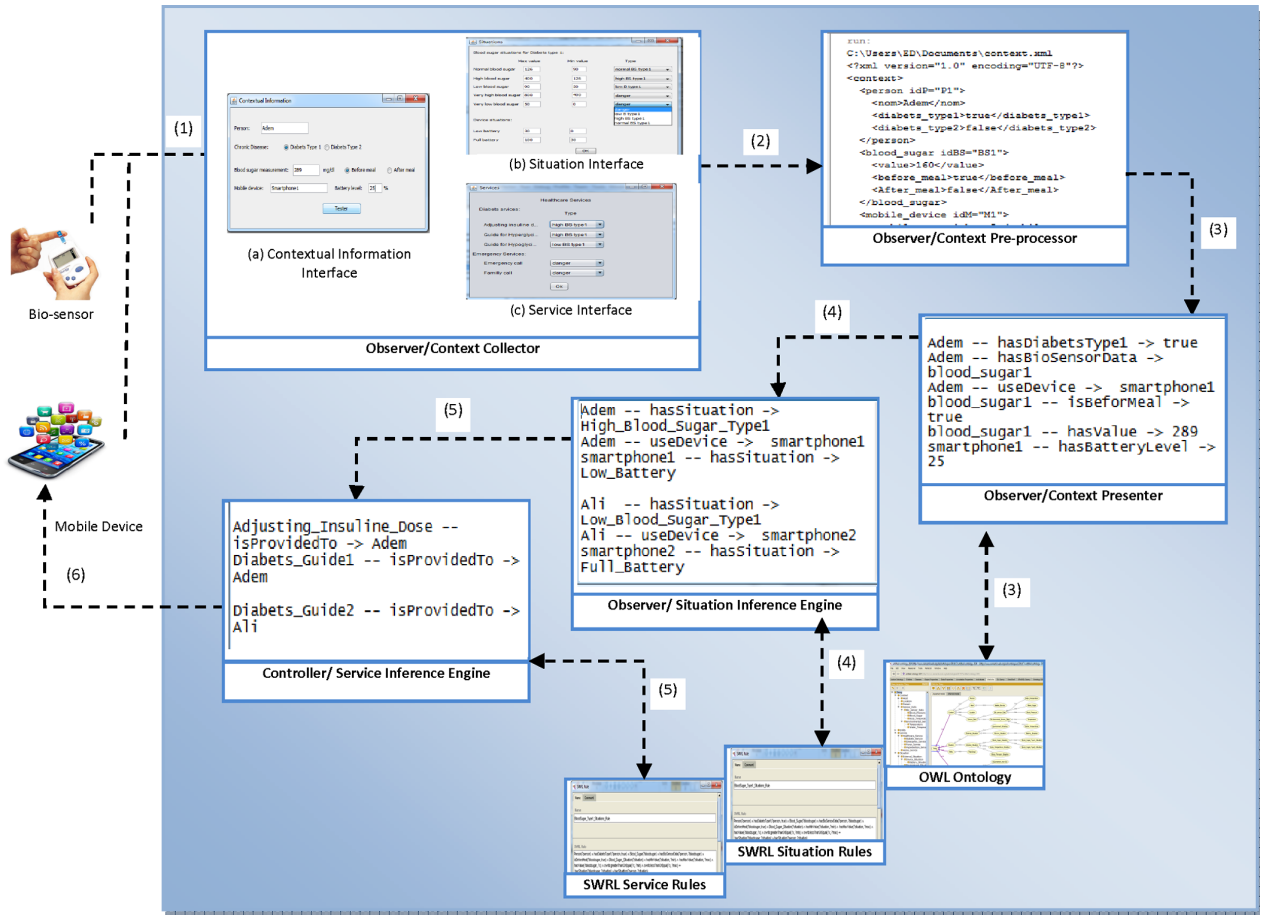


Table 5 Example of results

Person	*Blood sugar measurement* (ml/g)	*Mobile device's battery level* (%)	*Inferred health situation*	*Inferred healthcare service*	*Inferred device situation*
Adem	250	15	High blood sugar type1	Adjusting insulin dose	Low battery
Ali	68	52	Low blood sugar type1	Diabetes guide 2	Full battery
Alice	50	90	Very low blood sugar type1	Emergency call/Family call	Full battery
Anis	440	60	Very high blood sugar type1	Emergency call/Family call	Full battery
Iness	20	65	Very low blood sugar type1	Emergency call/Family call	Full battery
Karim	120	25	Normal blood sugar type 1	Diabetes guide 3	Low battery
Lilya	80	27	Low blood sugar type1	Diabetes guide 2	Low battery
Lina	480	68	Very high blood sugar type1	Emergency call/Family call	Full battery
Linda	55	35	Low blood sugar type1	Diabetes guide 2	Full battery
…	…	…	…	…	…

7 Discussion

With the progress of ubiquitous technologies, it is being possible to provide more smart and pervasive healthcare services in a smart environment. These services allow patients and elderly persons to live safely and independently in their own homes. The complexity of pervasive systems in general and U-Healthcare systems in particular is steadily increasing. In these systems, there is a growing variety of mobile computing devices, which are highly connective and can be used for different tasks in dynamically changing environments. Owing to this complexity, innovative ideas for the design and the development of this kind of systems must be found, as stated in Schmeck (2005); "*It is not the question whether self-organised and adaptive systems will arise but how they will be designed*". Furthermore, for providing the appropriate services to the user, those systems must be able to learn over time about not only the health measurements of patient, but also his current location, his profile, his activities, and his physical environment. Besides, for monitoring the health and wellbeing of patients and elderly persons anytime and anywhere, pervasive healthcare systems provide mobile services and applications that can be used on mobile devices. However, the obstacles related to the limited resources of mobile devices, such as battery lifetime, hinder the service continuity on this kind of systems. For that, device resources have to be taken into account during the execution of these systems, and the service continuity must be assured on mobile devices. Indeed, this kind of mobile systems which are related with their physical environment needs to adapt their behaviours according to the context change; in order to deal with the different resources presented in the environments. Those systems must be able to detect context information over time which comes from different and heterogeneous entities, deduce new situations from this information, and then adapt their behaviour automatically according to the deduced situations; in order to provide the appropriate services at the right time, in the right place and with the right manner. Consequently, we cannot disregard the importance of how the adaptive systems will be able to reason upon context information and adapt their behaviour according to the dynamic change of this context.

In this paper, we have proposed new approach that allows responding to both requirements of pervasive systems: How these systems will be designed and how they will be able to adapt to the dynamic change of context.

Our proposal is based on the combination of the observer/controller (O/C) architecture of organic computing paradigm with ontologies and rule-based approaches. This combination permits to get most of their benefits for realising a new autonomic adaptation approach for pervasive systems in general and pervasive healthcare systems in particular.

This approach allows keeping systems highly supervised and controlled by the observer/controller; detecting, analysing and reasoning upon useful contextual information using a hybrid reasoning engine based on the ontology and rules. Moreover, it allows adapting automatically the behaviour of system according to the current context in order to provide the appropriate service to the user.

As mentioned in the first section, several works have been proposed for adapting pervasive systems to the dynamic change of context. MUSIC (Pan et al., 2013; Romain et al., 2009) focus on adapting mobile application to the dynamic change of context (e.g., location, network connectivity) in order to satisfy the user requirements and device properties (battery, memory, CPU). Other works like Catarinucci et al. (2012), Kim and Chung (2014), Ko et al. (2006), and Zeshan and Mohamad (2012) proposed U-healthcare systems based on ontologies and rule-based reasoning. These works have proposed interesting ontologies that can be used in the ubiquitous healthcare filed. However, they are specified to the healthcare context-aware system and authors have modelled context according to their point of view. In addition, they have used rules for reasoning upon contextual information and deducing health situation without taking into account situations related to the mobile device's resources, which means that they have not taken into consideration the obstacles related to the limited resources of mobile devices; such as battery life time, and then they did not propose solutions to assure the service continuity on mobile devices.

Compared with those works, the combination of O/C architecture with ontologies and rule-based-approaches provides new design approach for context-aware healthcare systems that, in our knowledge, have not been proposed yet in this field. In addition, the context model that we have proposed allows not only representing and reasoning upon contextual information, but also providing a generic and flexible model to the developer that facilitates modelling context and developing context-aware system.

In our approach, context is represented using owl ontology. This ontology has three main classes (context class, situation class and service class) which are a concrete representation of generic concepts that can be used in any context-aware system. Our ontology is structured in three levels:

- contextual information level
- contextual situation level
- contextual service level.

The first level represents contextual information that can be used to deduce situations related to the user, where the second level represents different situations that can be used in the system. These situations can be *external situations* which represents generic situations (e.g., temperature situations and device resources situations), or *internal situations* which represents specific situations that can be specified by the domain expert; such as blood sugar situations. The representation of situations on ontology and, the separation between the contextual information and situations provide possibilities to add easily any situation needed in the system. Finally, the third level represents different services that can be provided by the system. Dividing the ontology on these three levels

facilitate the conception and the development of the context model.

On the other hand, for reasoning upon context, a set of SWRL rules are used. These rules are classified in two categories; *situation rules* and *service rules*. The first ones allow deducing automatically the current situations by inferring new properties between contextual information represented in the first level and situations represented in the second level of the ontology. As situations can be external or internal situations, situation rules also can be *generic* rules or *specific* rules (specified by the domain expert). *Generic rules* are used to deduce external situations, while specific rules are used to deduce internal situations. The *service rules* allow determining the appropriate service that must be provided to the user according to the deduced situation; by inferring new properties between the situations represented in the second level and the services represented in the third level.

Consequently, our model can be used in any context-aware system since we can represent generic and specific concepts and use generic and specific rules. We just need to change the specific concepts, internal situation, and make specific rules for a specific domain. Moreover, our model is scalable and extensible since we can easily add new contextual information and reasoning upon this new information. For instance, if a new sensor is added to the environment, like a heartbeat sensor for patients, we can add this information by just adding a sub-class of *bio-sensor-data* class, and creating new rules for this information. Furthermore, we have proposed to use mechanisms of dynamic service redeployments and reconfiguration (e.g., service migration, replacement, etc.) in order to assure the service continuity on mobile devices. This makes our approach more interesting since it allows adapting pervasive systems to the mobile device constraints such as battery lifetime.

8 Conclusion

Providing appropriate services to the user according to his current context is the main objective of pervasive healthcare systems; and assuring the continuity of these services is becoming necessary for this kind of systems. For that, they need to be supervised continually in order to detect context information; and adapt themselves automatically in response to the dynamic change of this context. The increasing complexity of pervasive healthcare systems requires finding new innovative ideas for designing and adapting these systems. In this paper, we have proposed new design approach for pervasive healthcare systems based on the observer/controller architecture, ontologies and rules. Our approach allows supervising the system, detecting useful contextual information, reasoning upon this information, and adapting the behaviour of system according to the current context by providing the appropriate service to the user. We have described in detail our architecture. It was shown how observer and controller are designed and how they can use ontologies and rule-based approaches. In addition, we proposed to use mechanisms of dynamic service redeployments when mobile devices will be exhausted; in order to assure the continuity of services. Actually, we have not developed our methods that allow reconfiguring and redeploying services from a device to other devices, but we propose to use existing works that are focusing on these mechanisms. In future works, we intend to extend the context model with new concepts and evaluate our architecture in more complex case study scenarios. Moreover, we will develop our mechanisms for the dynamic services reconfiguration and redeployment, including the migration of services from local devices to the cloud.

References

Agoston, T.C., Ueda, T. and Nishimura, Y. (2000) 'Pervasive computing in a networked world', *INET*, Japan, http://www.isoc.org/inet2000/cdproceedings/3a/3a_1.htm

Akyildiz, I.F., Su, W., Sankarasubramaniam, Y. and Cayirci, E. (2002) 'Wireless sensor networks: a survey', *Computer Networks. The International Journal of Computer and Telecommunications Networking*, Vol. 38, No. 4, pp.393–422.

Baldauf, M. and Dustdar, S. (2007) 'A survey on context-aware systems', *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 2, No. 4, pp.263–277.

Becker, B., Allerding, F., Reiner, U., Kahl, M., Richter, U., Pathmaperuma, D., Schmeck, H. and Leibfried, T. (2010) 'Decentralized energy-management to control smart-home architectures', *Architecture of Computing Systems – ARCS 2010*, Springer Berlin Heidelberg, pp.150–161.

Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A. and Riboni, D. (2010) 'A survey of context modelling and reasoning techniques', *Pervasive and Mobile Computing*, Vol. 6, No. 2, pp.161–180.

Branke, J., Mnif, M., Muller-Schloer, C. and Prothmann, H. (2006) 'Organic computing – addressing complexity by controlled self-organization', *The 2nd International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA 2006)*, Cyprus, pp.200–206.

Cabitza, F., De Paoli, F., Loregian, M. and Boselli, R. (2005) 'An adaptive middleware to support context-aware knowledge sharing', *The 25th IEEE International Conference on Distributed Computing Systems Workshops*, IEEE, pp.352–358.

Cassagnes, C., Roose, P., Dalmau, M. and Louberry, C. (2009) 'Kalimucho: software architecture for limited mobile devices', *Special Issue on the 2nd International Workshop on Adaptive and Reconfigurable Embedded Systems (APRES'09)*, Vol. 6, No. 3, pp.1–6.

Catarinucci, L., Colella, R., Esposito, A., Tarricone, L. and Zappatore, M. (2012) 'RFID sensor-tags feeding a context-aware rule-based healthcare monitoring system', *Journal of Medical Systems*, Vol. 36, No. 6, pp.3435–3449.

Chen, H., Finin, T. and Joshi, A.K. (2003) 'An ontology for context-aware pervasive computing environments', *The Knowledge Engineering Review*, Vol. 18, No. 3, pp.197–207.

Clark, K., Grove, M., Sirin, E., Pérez-Urbina, H., Klinov, P. and Rodríguez-Díaz, E. (2007) *Pellet: OWL 2 Reasoner for Java* [Online], Available at: http://clarkparsia.com/pellet (Accessed July, 2014).

Conversion of Glucose Values from mg/dl to mmol/l (2009) [Online], Available at: http://www.soc-bdr.org/rds/authors/unit_tables_conversions_and_genetic_dictionaries/conversion_glucose_mg_dl_to_mmol_l/index_en.html (Accessed 26 October, 2014).

Da, K., Dalmau, M. and Roose, P. (2014) 'Kalimucho: middleware for mobile applications', *The 29th Annual ACM Symposium on Applied Computing SAC '14*, ACM, Korea, pp.413–419.

Doukas, C. and Maglogiannis, I. (2012) 'Bringing IoT and cloud computing towards pervasive healthcare', *The Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, IEEE, Palermo, pp.922–926.

Doukas, C., Pliakas, T., Panayiotis, T. and Maglogiannis, I. (2012) 'Distributed management of pervasive healthcare data through cloud computing', *The Second International ICST Conference, MobiHealth 2011*, Springer Berlin Heidelberg, Kos Island, pp.386–393.

Eiter, T., Ianni, G., Krennwallner, T. and Polleres, A. (2008) 'Rules and ontologies for the semantic web', *Reasoning Web*, Springer-Verlag Berlin, pp.1–53.

Fensel, D. (2004). *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*, 2nd ed., Springer-Verlag plateforme, Berlin.

Henricksen, K., Indulska, J. and Rakotonirainy, A. (2002) 'Modeling context information in pervasive computing', *The First International Conference on Pervasive Computing*, Springer-Verlag, London, pp.167–180.

Henricksen, K., Indulska, J. and Rakotonirainy, A. (2003) 'Generating context management infrastructure From high-level context models', *The 4th International Conference on Mobile Data Management (MDM)-Industrial Track*, pp.1–6.

Horridge, M., Musen, M., Nyulas, C., Tu, S. and Tudorache, T. (n.d.) *Protégé* [Online], Available at: http://protegewiki.stanford.edu/wiki/Main_Page (Accessed February, 2014).

Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B. and Dean, M. (2004) *SWRL: A Semantic Web Rule Language Combining OWL and RuleML* [Online], Available at: http://www.w3.org/Submission/SWRL/ (Accessed July, 2014).

Ireson-Paine, J. (1996) *Rule-based Systems* [Online], Available at: http://www.j-paine.org/students/lectures/lect3/lect3.html (Accessed June, 2014).

Keling, D. (2014) *Kalimucho – A: Autonomic Knowledge – Based Context-Driven*, Université De Pau et Des Pays De L'Adour, Pau, France.

Kephart, J.O. and Chess, D.M. (2003) 'The vision of autonomic computing', *IEEE*, Vol. 36, No. 1, pp.41–50.

Kim, J. and Chung, K-Y. (2014) 'Ontology-based healthcare context information model to implement ubiquitous environment', *Multimedia Tools and Applications*, Vol. 71, No. 2, pp.873–888.

Kim, J., Kim, J., Lee, D. and Chung, K-Y. (2014) 'Ontology driven interactive healthcare with wearable sensors', *Multimedia Tools and Applications*, Vol. 71, No. 2, pp.827–841.

Kluge, F., Uhrig, S. and Ungerer, T. (2008) 'An operating system architecture for organic computing in embedded real-time systems', *Autonomic and Trusted Computing*, Springer Berlin Heidelberg, pp.343–357.

Ko, E-J., Eee, H-J. and Eee, J-W. (2006) 'Ontology-based context-aware service engine for U-healthcare', *The 8th International Conference on Advanced Communication Technology ICACT 2006*, IEEE, Phoenix Park, pp.632–637.

Lee, C., Jian, Z. and Huang, L. (2005) 'A fuzzy ontology and its application to news summarization', *IEEE Trans. Syst. Man. Cybern.*, Vol. 35, No. 5, pp.859–880.

Lee, H. and Kwon, J. (2013) 'Ontology model-based situation and socially-aware health care service in a smart home environment', *International Journal of Smart Home*, Vol. 7, No. 5, pp.239–250.

Louberry, C. (2010) *Kalimucho: Adaptation au Contexte pour la Gestion de la Qualité de service*, PhD Thesis, Ecole Doctorale Des Sciences Exactes Et De Leurs Applications, France.

Louberry, C., Roose, P. and Dalmau, M. (2011) 'Kalimucho: Plateforme d'Adaptation des Applications Mobiles', *Conférence Internationale sur les NOuvelles Technologies de la REpartition (NOTERE 2011)*, Paris, pp.83–90.

Orgun, B. and Vu, J. (2006) 'Ontology and mobile agents for interoperability in heterogeneous medical information systems', *Comput. Biol. Med.*, Vol. 36, No. 7, pp.817–836.

Pan, B., Wang, X., Song, E., Lai, C-F. and Chen, M. (2013) 'CAMSPF: cloud-assisted mobile service provision framework supporting personalized user demands in pervasive computing environment', *9th International in Wireless Communications and Mobile Computing Conference (IWCMC)*, IEEE, Sardinia, pp.649–654.

Richter, U. and Christian, M. (2006) 'Towards a generic observer/controller architecture for organic computing', *Informatik 2006 – Informatik fur Menschen!*, pp.112–119.

Rolim, C.O., Rolim, C.O., Koch, F.L., Westphall, C.B., Werner, J., Fracalossi, A. and Salvador, G.S. (2010) 'A cloud computing solution for patient's data collection in health care institutions', *The Second International Conference on eHealth, Telemedicine, and Social Medicine, eTELEMED 2010*, IEEE, St. Maarten, pp.95–99.

Romain, R., Paolo, B., Yun, D., Frank, E., Svein, H., Jorge, L., Alessandro, M. and Ulrich, S. (2009) 'Music: middleware support for self-adaptation in ubiquitous and service-oriented environments', *Software Engineering for Self-Adaptive Systems*, Springer Berlin Heidelberg, pp.164–182.

Rosati, R. (2006) 'Integrating ontologies and rules : semantic and computational issues', *Reasoning Web*, Springer Berlin Heidelberg Edition, Springer-Verlag Berlin Heidelberg, pp.128–151.

Schloer, M. (2004) 'Organic computing: on the feasibility of controlled emergence', *The 2nd IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, pp.2–5.

Schmeck, H. (2005) 'Organic computing – a new vision for distributed embedded systems', *The 8th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2005)*, Comput I., ed., Los Alamitos, pp.201–203.

Schmeck, H., Müller-Schloer, C., Çakar, E., Mnif, M. and Richter, U. (2011) 'Adaptivity and self-organisation in organic computing systems', *Organic Computing – A Paradigm Shift for Complex Systems*, Springer Basel, pp.5–37.

Sheng, Q.Z. and Benatallah, B. (2005) 'ContextUML: a UML-based modeling language for model-driven development of context-aware web services', *International Conference on Mobile Business ICMB'05*, IEEE, pp.2006–2012.

Smith, M., Welty, C. and McGuinness, D. (2004) *Owl Web Ontology Language Guide* [online], Available at: http://www.w3.org/TR/owl-guide/ (Accessed February, 2014).

Strang, T. and Linnhoff-Popien, C. (2004) 'A context modeling survey', *The First International Workshop on Advanced Context Modelling, Reasoning and Management UbiComp 2004*, Nottingham.

Tomforde, S., Cakar, E. and Hähner, J. (2009) 'Dynamic control of network protocols – a new vision for future self-organised networks', *The 6th Int. Conf. on Informatics in Control, Automation, and Robotics (ICINCO'09)*, pp.285–290.

Tomforde, S., Prothmann, H., Branke, J., Hähner, J., Mnif, M., Müller-Schloer, C., Richter, U. and Schmeck, H. (2011) *Observation and Control of Organic Systems, Organic Computing – A Paradigm Shift for Complex Systems, Autonomic Systems*.

Uschold, M. and Jasper, R. (1999) 'A framework for understanding and classifying ontology applications', *The IJCAI-99 Workshop on Ontologies and Problem-Solving Methods*, Stockholm.

W3C (1999) *Composite Capability/Preference Profiles (CC/PP)* [Online], Available at: http://www.w3.org/TR/NOTE-CCPP/ (Accessed February, 2014).

W3C (2013) *Inference* [Online], Available at: http://www.w3.org/standards/semanticweb/inference (Accessed Jun 2014).

Wang, M-H., Lee, C-S., Hsieh, K-L., Hsu, C-Y., Acampora, G. and Chang, C-C. (2010) 'Ontology-based multi-agents for intelligent healthcare applications', *Journal of Ambient Intelligence and Humanized Computing*, Vol. 1, No. 2, pp.111–131.

WAPFORUM (2001) *User Agent Profile (UAProf)* [Online], Available at: http://www.wapforum.org (Accessed February 2014).

Weiser, M. (1991) 'The computer of the twenty-first century', *Scientific American*, Vol. 265, No. 3, pp.94–100.

Ye, J., Coyle, L., Dobson, S. and Nixon, P. (2007) 'Using situation lattices to model and reason about contex', *The Workshop of Modeling and Reasoning Context (MRC 2007)*, Roskilde, Denmark, pp.1–12.

Yuriyama, M. and Kushida, T. (2010) 'Sensor-cloud infrastructure: physical sensor management with virtualized sensors on cloud computing', *The 13th International Conference on Network-Based Information Systems*, IEEE, Takayama, pp.1–8.

Zeshan, F. and Mohamad, R. (2012) 'Medical ontology in the dynamic healthcare environment', *Procedia Computer Science*, Vol. 10, pp.1–1228.