A Framework for Anomaly Diagnosis in Smart Homes Based on Ontology

Etienne Pardo, David Espes, Philippe Le Parc

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 00 (2016) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

The 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016)

A Framework for Anomaly Diagnosis in Smart Homes Based on Ontology

Etienne Pardo[a], David Espes[a,*], Philippe Le-Parc[a]

[a]*Université de Bretagne Occidentale - LabSTICC, 20 avenue Le Gorgeu, 29200 Brest, France*

Abstract

Smart homes are pervasive environments to enhance the comfort, the security, the safety and the energy consumption of the residence. An ambient intelligence system uses information of devices to represent the context of the home and its residents. Based on a context database, this system infer the daily life activities of the resident. Hence, abnormal behavior or chronic disease can be detected by the system. Due to the complexity of these systems, a large variety of anomalies may occur and disrupt the functioning of critical and essential applications. To detect anomalies and take appropriate measures, an anomaly management system has to be integrated in the overall architecture. In this paper, we propose an anomaly management framework for smart homes. This framework eases the work of designers in the conception of anomaly detection modules and processes to respond to an anomaly appropriately. Our framework can be used in all heterogeneous environments such as smart home because it uses Semantic Web ontologies to represent anomaly information. Our framework can be useful to detect hardware, software, network, operator and context faults. To test the efficiency of our anomaly management framework, we integrate it in the universAAL middleware. Based on a reasoner, our framework can easily infer some context anomalies and take appropriate measures to restore the system in a full functioning state.

Peer-review under responsibility of the Conference Program Chairs.

Keywords: Context-aware Framework ; Anomaly management ; AAL ; Smart home ; Ontology

1. Introduction

Smart homes refer to residences equipped with technology to monitor the environment and provide advance services based on the context of daily life activities of the inhabitants. Devices (i.e., sensors, actuators, computers...) are scattered everywhere in the residence and an ambient intelligence system uses the information they send to offer a better quality of life. The ambient intelligence system proposes automated appliance control and assistive services to the residents. Hence, smart homes enhance personal comfort, security, safety and energy consumption [1].

However, smart homes are composed of heterogeneous networks and device's capacities [2]. Devices differ in terms of both communication technologies and capabilities (software and hardware). Indeed, these devices often use dif-

* Corresponding author. Tel.: +33-2-98-01-83-06 ; fax: +33-2-98-01-80-11.
E-mail address: david.espes@univ-brest.fr

Peer-review under responsibility of the Conference Program Chairs.

ferent communication technologies where the interoperability cannot be ensure. Moreover, these devices range from high-end PC devices to low-end battery-less devices. The heterogeneity of networks and device's capacities increase the complexity to exchange information between them. The ambient intelligence system need a same representation of the information to monitor the context appropriately.

Semantic Web technologies such as Web Ontology Language (OWL) are good candidates to provide data interoperability between these devices. Semantic Web ontologies can be used to classify information and formally describe concepts. Indeed, ontologies describe the relations between objects that represent the domain of interest. Hence, ontologies increase the inferring power of the ambient intelligence systems. Due to the limitless interoperability possibilities proposed by Semantic Web ontologies, researchers propose context-aware middlewares [10 11], frameworks [3 4 5 6 7] and architectures [8 9] in order to propose a generic platform to ease the development of applications based on the context of daily life activities of the residents. These researches can increase the comfort and safety of the inhabitants while enhancing the energy consumption of the residence.

Smart homes can propose a large variety of applications. One of the most important application is the health care of residents [12]. People ageing occurs in every country all over the world. Indeed, the expected population of the EU-28 around 2050 will reach 525.5 million inhabitants while 28% will be over 65 years and 11% over 80 [13]. Population ageing raises the dependency level of the elder people. Smart home is convenient to monitor the health of dependent people and offer healthcare services remotely. In the same way as for generic smart home infrastructure, context-aware middlewares [14 15], frameworks [16 17 18] and architectures [19 20] are proposed to manage the health of the residents. All these approaches use Semantic Web ontologies such as OWL to ensure interoperability of information and infer the daily activities of the residents. These methods can detect changes in the behavior of residents or chronic diseases and inform caregivers of health problems.

Due to the large variety of applications, anomalies in smart homes can put at risk the life of the residents. For example, caregivers cannot detect chronic diseases if the sensors worn by the residents send false information. In the same way, in case of a broken human fall detection sensor, emergency workers may be informed about the problem after a long time while causing an excessive stress or increasing the suffering of the fallen resident. Hence, an anomaly management system becomes an indispensable part of the overall architecture.

In this paper, we propose an context-aware anomaly management framework. Our framework uses the semantic web ontologies to represent the anomaly information. So, our framework can be uses in all types of heterogeneous environments such as smart homes. Our framework eases the design of anomaly detection modules and anomaly management services. it incorporates a reasoner to infer misbehavior or context problems. Our framework is designed to manage hardware, software, network, operator and context faults.

The remainder of this paper is organized as follow. We present in section II, a fault model for smart homes and the design of the anomaly ontology. In section III, we present our anomaly management framework. In section IV we provide the details of its integration in the universAAL middleware. Finally, we conclude with some insights and related perspectives, in section V.

2. Smart home anomaly

Before delving into the smart home related faults, it matters to define what a fault is. Fault, failure, error... are many words used to refer to a system in an undesired configuration. Instead of classifying these words in a hierarchy, the higher level concept of anomaly will be used. An anomaly is an abnormal, or unexpected, situation or behavior of a part of a system. This is a "should not/never happen" phenomenon.

2.1. New anomaly model

From "Multi Agents Systems" (MAS) to "Service Oriented Architecture" (SOA), Distributed Systems vary in forms and aspects / concepts. As with anything, distributed systems are not perfect, and present some troubles. Whether they are human controlled or not, potential anomalies may originate from their conception, their usage, or malevolence. An anomaly in Distributed Systems can be of four types [21]: hardware anomaly, software anomaly, network anomaly and operator anomaly. Hardware anomalies concern the problems encountered by physical devices. Software anomalies concern the problems encountered by the logical program executed on the physical device. Net-

work anomalies occur when the communication between at least two devices fails or is degraded. Operator anomalies are errors related to human factor and may be made during system design, manufacturing, implementation, operation and maintenance.

This description does not reflect all the possibilities that are of concern with pervasive technologies. For example, when a furnace breaks, an hardware fault is expected. However, when a smoke-detector detects smoke, this is not a fault in itself. The device works properly. Still, the smoke is not expected to be present, so the situation is an anomaly.

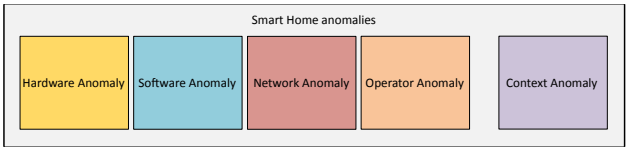

Fig. 1. Smart home anomaly

To this end, the extension from Fig. 1 is suggested. The context anomaly expresses the various alarms that may trigger due to ”good work but bad situation”. Unlike the previous anomalies (Hardware, Software, Network and Operator), this one helps to express more various situations.

The case of malicious act perpetrated through or to the system is deliberately left outside this scope. They are indeed a real threat, specifically for smart homes. However, they can be translated as alarms raised by policy-check softwares, or as previous categories from Fig. 1.

In smart homes, the good execution of the system should include the safety of the users and/or assisted people. This goal implies that some ”correct behaviors” (such as alarms. . .) must be treated as anomaly. For example, there is no reason to treat differently[1] a fire alarm than a burnt-out light notice. Hence the addition in Fig. 1.

In this context, an ontology has been built to tackle anomaly expression in smart homes. Due to the wide variety of situations, and elements of the system, the ontology is conceived as a high-level abstraction and cut down in 3 parts.

1. An ontology describing the components; the various part of a system and its relevant surrounding. This part is detailed in section 2.2.
2. An ontology describing the anomaly; centered mainly on the expressed knowledge of the anomaly occurring or suspected to occur This part is detailed in section 2.3.
3. An ontology describing the communication messages about the anomaly. This part is detailed in section 2.4.

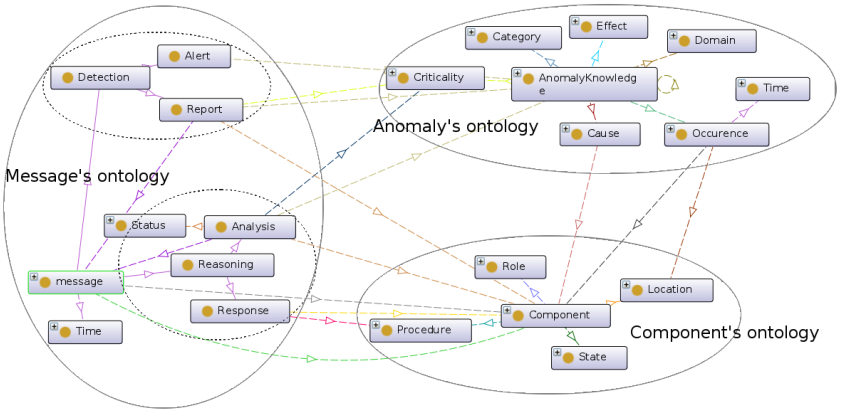

Fig. 2. Anomaly’s ontology overview. The various parts are surrounded. Detection (2.4.1) and Reasoning (2.4.2) aspect are put into light.

[1] except for the potential severity

2.2. Component

A component is a part of the system; the system being what is studied or used, with its relevant dependencies. By itself, a component can be studied as a system, with underlying components. For instance, a basic smart home's system could be: the devices (sensors, actuators...); the network; the home itself (including its furniture); the dependent people and any other inhabitant. From Fig. 2, a component has various properties.

- An identification key that permits to uniquely identify a given component.
- A role that defines what can be expected of the component. Expected roles are:
  - a node, either an actuator, a sensor, a network relay or a computing device;
  - an authority, defining the type of rights granted to the component;
  - a resource, either a consumable that can be used or an alterable medium.
- A location that refers to some system's known areas.
- A state that defines both the component's Finite State Machine's state (on, off, etc.), and how well does it perform (is it broken?).

2.3. Anomaly

An anomaly is an "observed, abnormal behaviorof (a part of) a system". An anomaly's description should then express, at least, which part of the system is affected, by what. Ideally, this should be completed with additional information, such as which components observe the anomaly, or the worth of its management. As proposed in Fig. 2, an anomaly can be described by:

- a set of effects. Each effect describes one aspect of the observed phenomenon;
- a set of causes. Each cause refers to a component suspected to be the source of the anomaly;
- a set of occurrences. Each occurrence refers to where and when which component observed the anomaly;
- a criticality. It is a time dependent worth attributed to the anomaly management;
- a set of domains. Each domain relates to a specific kind of consequence of the anomaly;
- a category. An anomaly can happen once (transient), periodically (recurrent) or definitively (permanent).

An anomaly may be linked to other anomalies it may be related to. These other anomalies may be suspected occurrences of the same anomaly. They may be improved knowledge of the situation. Or these anomalies may be unrelated, and are wrongly linked together.

2.4. Anomaly message communication ontology

Once an anomaly is detected, the system must be warned of its occurrence. Then, additional information can be gathered and processed. Finally, appropriated actions to mitigate its effect can be suggested. These 4 steps can be classified into 2 categories: Detection and Reasoning. The former is about anomaly acknowledgment,whereas the latter is about anomaly management. When an anomaly is detected, an anomaly message is sent. Once received, a component infers a suited reaction. If unable to process the message, it transmits the message to a more suited component.

2.4.1. Detection

The detection part of the message ontology focuses on the knowledge discovery. This relies on components' survey to trigger a warning about abnormalities. Such triggered warnings fall into 2 categories: the alert, when the abnormality is firstly discovered; the report, when the knowledge about the abnormality is completed.

Alert. The alert is a message which warns the system an anomaly has been detected. It is completed by an instance of the anomaly ontology, with as much information as available from the emitter. Thus, the provided anomaly may be incomplete, or even almost empty.The alert is emitted by any component able to detect the occurrence of an anomaly.

Report. The report is a message which completes previous knowledge about the related anomaly. The information it provides is not definitive, as it can be erroneous. Though, the report should contain as reliable as possible information. The report is emitted by any component which knows of an anomaly, and can provide complementary information.

2.4.2. Reasoning

The reasoning part of the message ontology focuses on the improvement of the knowledge about the anomaly. This relies on the component's ability to reason, and to process information. When an alert or a report occurs, two reactions are expected of the system: the analysis, when the system improves and strengthens the knowledge of the anomaly; the response, when the system tries to mitigate or solve the anomaly.

Analysis. Similarly to the report, the analysis completes previous knowledge about the anomaly. The distinction is that an analysis provide, if not definitive, up to date and reliable information. Only authorized components can emit an analysis.

Response. The response is a (complex) instruction sent to mitigate the anomaly, to the point of resolving it if possible. It consists in a pseudo-program; an ordered tree of procedures to perform, sent to the relevant components. The exact content of the response depends on the concerned components; each component understanding a set of procedures. The reasoner might require to adapt an ideal resolution / mitigation to what is available. The response can only be emitted by authorized, well suited components.

Anomaly messages can be published by any components. According to the inferring capability of the component, a report may also be published. Unlike anomaly messages, reasoning related messages can only be published by authorized components. Because of their implication in the outcome of the system, analysis and responses must be given great care.

3. Context-aware framework for anomaly management

During the realization of a project, it is time consuming to manage anomalies. More so as many frameworks or middlewares manage only a subset of the expected ones. Hence, it is necessary to have abstract method to manage them.

Our framework eases the developers' work related to anomaly management. It offers a high level interface to trigger anomaly related messages that can be sent to specific components/services or to all of them. Through these messages, the system can determine if it fails due to transient or persistent anomalies and react accordingly. One of the main benefit of our framework is hardware- and software-independence. Indeed, the interoperability is ensured by the use of Semantic Web ontology to define an anomaly. Accordingly to the smart home ontology we proposed (cf. Section 2), five types of anomaly can be managed: hardware anomaly, software anomaly, network anomaly, operator anomaly and context anomaly.

Our framework is given in Fig. 3.a. It is composed of five anomaly detection modules (ADM). They extract from the context, including other anomalies, their relevant occurring anomalies. The devices ADM manages the hardware failures and related anomalies. The services ADM manages the software failures. The network ADM manages the anomalies related to the communication layer. The application ADM manages part of the third-parties' software failures. It overlaps both software and operator anomalies from Fig. 1. On the other hand, the context anomalies are fused inside the context manager that is usually proposed by context-aware frameworks or middlewares. The context manager (situation publisher, semantic reasoner, knowledge core and context aggregator) is enhanced to manage anomaly as any other context event. Though only the context manager is detailed, each anomaly detection module may have their own reasoner, knowledge core, and so on.

Our anomaly management framework is generic enough to be integrated into context-aware and legacy frameworks or middlewares. Hence, our proposal with the anomaly ontology can extend existing context-aware frameworks or middlewares, to take into account anomalies. The same way context-aware frameworks or middlewares ease the context management for the applications, Anomaly-aware framework would ease the management of anomaly.

Fig. 3.b presents our framework, integrated into a context aware middleware. The modifications are quite light. The different ADMs from Fig.3a are placed in their expected place: their respective manager in the middleware; or

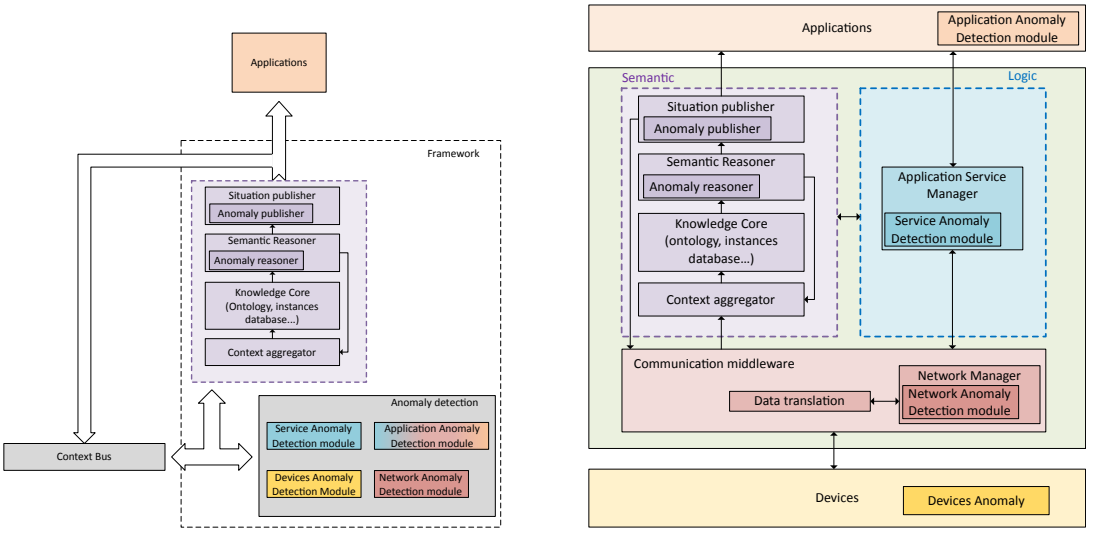

Fig. 3. a) At the left, the anomaly management framework. b) At the right, Framework integration to extend legacy middleware systems to support anomaly management

their related component. In service oriented, context-aware middlewares, the focus is given to communication: service and context are transmitted to and from the applications. Though in Fig.3b, only the context manager is detailed, the service manager is similar. Both have their own reasoner, publisher, and so on. Both managers interact to improve their own efficiency, as well as the overall efficiency.

4. Integration in universAAL

4.1. Brief description of universAAL

Our framework is generic enough to be incorporated into smart home middlewares with little work. Due to its specific design for AAL systems, UniversAAL middleware [14] (uAAL) is selected for the implementation.

The UniversAAL middleware is an EU FP7 project middleware geared toward AAL and related tasks. Following of a few previous (FP6) projects, uAAL aims to be a relevant standard in the AAL middlewares world. This service oriented architecture (SOA) is written in Java 6 and relies on OSGi for the deployment. The applications register themselves to the middleware and expose their interfaces as ontology instance's patterns. In this middleware, the communication layers are abstracted, and applications receive only the messages for which they registered. Changes are expressed as context events, and operations are translated into service requests, and both are ontology's instances.

4.2. UniversAAL class diagram

Fig. 4.a provides an overview of the uAAL's organization.Some of the parts are optional. This is the case for the "security" and its subclasses, as well as the situation reasoner. These parts are then easier to make evolve. As a side bonus, is eases the development of unrelated services.

In Fig. 4.b, the additions to uAAL are represented by gray classes.Most additional services (publisher and aggregator) are not fully integrated with uAAL yet. Hence, they are just applications.

Since the potential anomalies are dependent of the applications and the system, managing the detection of all and any anomaly is almost an impossible task. Therefore, the universAAL middleware manages only a small subset of the anomaly detection, centered on the communication. For this reason, specific detection applications are built. They take care of basic aspects of the life cycle of the associated applications. These "application managers" are jointly configured with a basic rule-based expert system. Current "proof-of-concept" anomaly responses are start/stop specific instance of a service or specific service.

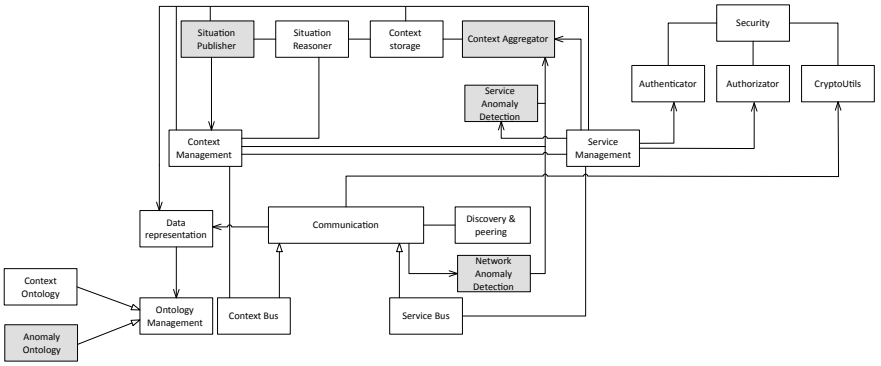

Fig. 4. a) UniversAAL class diagram (white classes). b) UniversAAL class diagram extension for modeling fault management (gray classes)

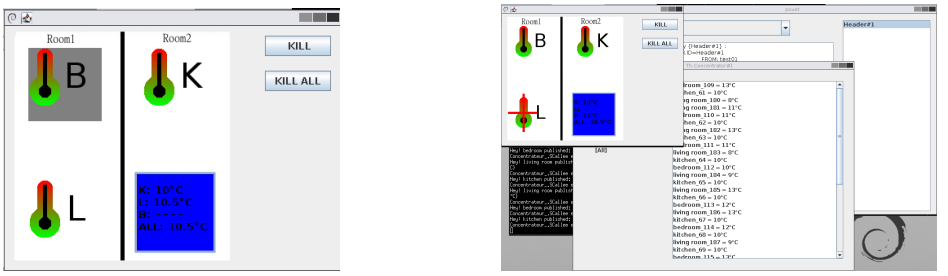

Fig. 5. a) At the left, the main interface (with one of the bedroom sensors momentarily disabled). b) At the right, the full interface (just after the reactivation of the bedroom sensor).

4.3. Results

Illustrated by Fig. 5.a, a proof of concept has been implemented in uAAL. The temperature is monitored in two rooms: Room1 and Room2. Room1 is equipped with two temperature sensors (B and L) whereas Room2 is equipped with a temperature sensor (K) and a temperature regulator (the blue square). The temperature regulator has two processes: one to regulate the temperature of the rooms and one to manage the anomalies. Only one temperature sensor is active at a time in each room, so the temperature sensor B is in a sleeping mode.

Periodically, each active temperature sensor sends the temperature of the room to the regulator. The regulator adjusts the heat of the radiator according to the temperature requested by the resident.

In this test UI, by clicking on the desired sensor, the user requests the termination of the sensor. In Fig. 5.b, the user disables the temperature sensor L. The regulator stops to receive temperature messages from the sensor L. After a defined waiting time, the regulator knows that the sensor L is down. It sends an anomaly message to the anomaly manager. When the anomaly manager receives this message, it infers that the sensor L is down. It sends a message to the sensor B in order to wake it up, while the regulator monitors the temperature of the room Room1.

5. Conclusion

Smart homes use heterogeneous networks and device's capacities to offer a better quality of life to the residents and provide assistive services. In the literature, a lot of proposals have been realized to consider the user context and infer the activity performed by the users. To ensure data interoperability between the devices, these proposals use Semantic Web ontology.

However, they do not take into account the anomalies that a system may encounter. Due to specific applications for smart homes such as healthcare, the potential failures of the system have to be taken into consideration. Hence, anomaly management systems become an indispensable part of the overall architecture.

We propose in this paper an anomaly management framework to ease the design of applications and anomaly detection modules. By using such a framework, the designer of the system can easily define specific anomalies and some rules to react at some inappropriate changes in the system. Our framework provides a high level of detail in anomaly conditions. Through the anomaly messages, the system can determine if it fails for reasons such as transient anomaly or persistent anomaly and infer the actions to restore the system in a functioning state or mitigate the anomaly. To ensure interoperability, our framework is based on Semantic Web ontology to describe an anomaly (represented by a set of objects and the interactions between them). The ontology can describe five types of anomaly: hardware anomaly, software anomaly, network anomaly, operator anomaly and context anomaly.

The framework has been integrated in the context-aware middleware universAAL. This proof of concept shows the effectiveness of our framework. Indeed, it eases the development of specific applications related to the anomaly management. Developers can easily represent an anomaly through its ontology or create applications that trigger anomalies when an unexpected event occurs. Indeed, they only need to use high level interface to trigger an anomaly.

References

1. Alam MR, Reaz MBI, Ali MAM. A Review of Smart Homes - Past, Present, and Future. *IEEE Transactions on Systems, Man, and Cybernetics* 2012. **42**(6):1190-1203.
2. Rossi L, Belli A, De Santis A, Diamantini C. Interoperability issues among smart home technological frameworks. *10th IEEE International Conference on Mechatronic and Embedded Systems and Applications (MESA)* 2014.
3. Liang Y, Zhou X, Yu Z, Wang H, Guo B. A Context-Aware Resource Management Framework for Smart Homes. *5th International Conference on Ubiquitous Information Technologies and Applications (CUTE)* 2010.
4. Sohn M, Jeong S, Lee HJ. Self-Evolved Ontology-Based Service Personalization Framework for Disabled Users in Smart Home Environment. *Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)* 2013.
5. Uhm Y, Hwang Z, Lee M, Kim Y, Kim G, Park S. A Context-Aware Multi-Agent System for Building Intelligent Services by the Classification of Rule and Ontology in a Smart Home. *32nd IEEE Conference on Local Computer Networks* 2007.
6. Grassi M, Nucci M, Piazza F. Towards an ontology framework for intelligent smart home management and energy saving. *IEEE International Symposium on Industrial Electronics (ISIE)* 2011.
7. Cheong Y-G, Kim Y-J, Yoo SY, Lee H, Lee S, Chae SC, Choi HJ. An ontology-based reasoning approach towards energy-aware smart homes. *IEEE Consumer Communications and Networking Conference (CCNC)* 2011.
8. Reinisch C, Kofler MJ, Kastner W. ThinkHome: A smart home as digital ecosystem. *4th IEEE International Conference on Digital Ecosystems and Technologies (DEST)* 2010.
9. Bonino D, Castellina E, Corno F. DOG: An Ontology-Powered OSGi Domotic Gateway. *20th IEEE International Conference on Tools with Artificial Intelligence* 2008.
10. Singh J, Hassanzadeh N, Rea S, Pesch D. Semantics-empowered middleware implementation for home ecosystem gateway. *IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)* 2014.
11. Hoque MR, Kabir MH, Jang J-H, Yang S-H. Middleware aided context-aware service for smart home. *18th IEEE International Symposium on Consumer Electronics (ISCE 2014)* 2014.
12. Chan M, Estve D, Escriba C, Campo E. A review of smart homes - present state and future challenges. *Comput Methods Programs Biomed* 2008. **91**(1):55-81.
13. http://ec.europa.eu/eurostat/statistics-explained/index.php/Population_structure_and_ageing (accessed on January 30, 2016).
14. http://www.universaal.org (accessed on January 30, 2016).
15. Evchina Y, Dvoryanchikova A, Lastra JLM. Ontological framework of context-aware and reasoning middleware for smart homes with health and social services. *IEEE International Conference on Systems, Man, and Cybernetics (SMC)* 2012.
16. Hennessy M, Oentojo C, Ray S. A framework and ontology for mobile sensor platforms in home health management. *1st International Workshop on the Engineering of Mobile-Enabled Systems (MOBS)* 2013.
17. Valero MA, Vadillo L, Penalver A et al. An Implementation Framework for Smart Home Telecare Services. *Future Generation Communication and Networking (FGCN 2007)* 2007. **2**:60-65.
18. Kang D-O, Kang K., Lee H-J, et al. A Systematic Design Tool of Context Aware System for Ubiquitous Healthcare Service in a Smart Home. *Future Generation Communication and Networking (FGCN 2007)* 2007. **2**:49-54.
19. Kim J, Choi H-S, Wang H. POSTECH's U-Health Smart Home for elderly monitoring and support. *IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM)* 2010.
20. Kuusik A, Reilent E, Loobas I, Parve M. Software architecture for modern telehome care systems. *6th International Conference on Networked Computing (INC)* 2010.
21. Bruning S, Weissleder S, Malek M. A Fault Taxonomy for Service-Oriented Architecture. *10th IEEE High Assurance Systems Engineering Symposium (HASE)* 2007.