International Journal of Smart Home
Vol. 10, No. 5 (2016), pp. 39-54
http://dx.doi.org/10.14257/ijsh.2016.10.5.05

Agent-Based Context-Aware Architecture for a Smart Living Room

Moeiz Miraoui[1], Sherif El-etriby[2], Abdulasit Zaid Abid[3]and Chakib Tadj[4]

[1,3]Al-Leith Computer College, Umm Al-Qura University, KSA
[2]Faculty of Computers and Information, Menoufia University, Egypt
[4]Ecole de Technologie Supérieure, University of Québec, Canada
[1]mfmiraoui@uqu.edu.sa, [2]Sherif.el-etriby@ci.menofia.edu.eg,
[3]azabid@uqu.edu.sa, [4]Chakib.Tadj@etsmtl.ca

Abstract

Technology is moving beyond the personal computer towards a trend of embedded microprocessors in everyday objects and home appliances. The recent advances in sensor networks and devices with computing ability have provided us the necessary technology to make smart spaces. In such spaces, user activity and behavior are taken into account in order to provide adequate and accurate adapted services to the current context. Services are provided proactively (without explicit user intervention) and in an unobtrusive manner. The main objective of smart spaces is to provide services to the user for improved comfort, energy savings, security, and tremendous benefits for elderly persons living alone or persons with disabilities. Despite the interesting number of proposed architecture for building smart spaces, there still exists a lack of a generic software architecture for the development of such spaces. The major weakness of proposed architecture is that they have not dealt in depth with context-awareness, which is a key feature especially in context-aware services adaptation in smart spaces. In this paper, we propose a multi-agent architecture for building a smart living room with a focus on context-awareness aspects. The proposed architecture is generic enough to be easily used in any smart space.

Keywords: *Smart space, Architecture, Agent, Context-awareness, Service, Adaptation*

1. Introduction

Recent advances in sensor network communication technologies and devices with computing ability have brought us towards the era of pervasive and ubiquitous computing. Devices in pervasive environments dynamically and proactively (without explicit user intervention) adapt their behavior to current context changes in order to provide adequate services to the user. Research on smart spaces typically falls under the banner of pervasive and ubiquitous computing, which tries to move from spaces filled with smart devices and appliances to smart spaces. So far, there is no explicit or common definition of smart spaces, which are also known as Ambient Intelligence or Ambient Assisted Living. The most useful definition of such spaces is the one proposed by D. J. Cook and S. Das [1]: "Smart space is able to acquire and apply knowledge about its environment and to adapt to its inhabitants in order to improve their experience in that environment". Smart spaces have been an emerging issue in academic research and research in these areas, and smart home automation is especially growing rapidly in the last years. Smart spaces must be capable of changing their behavior dynamically based on users' activities and the environment. Devices and appliances in a smart space are capable of interacting with people in order to provide intelligent services to users for improved comfort (*i.e.*, quality of life), energy saving, security, and tremendous benefits for an elderly person living alone to increase their level of autonomy. The development of smart spaces can be

ISSN: 1975-4094 IJSH

difficult and costly without the adequate support of a computing architecture, which is considered to be the first major challenge. An important number of smart space architectures, middleware, and frameworks have been developed in the previous years with diverse techniques. However, there still lacks a general model to guide the system design. In addition, major shortcomings of these systems are weak in supporting context-awareness, which is a key feature in such spaces and greatly helps in the adaptation services task. Some proposed architecture focus on one aspect (*e.g.*, physical layer) and neglect other important aspects (*e.g.* context-aware services adaptation), which limit their usability to a specific cases of smart spaces. Our aim is to propose a multi-agent software architecture for building a smart living room. We tackle in depth the context-awareness aspect by focusing on context definition and modeling. We propose an architecture that contains the essential modules needed in a smart space in order to overcome some weakness of previous works.

This paper is organized as follows: Section 2 presents an enlarged survey of related work. Section 3 discusses weakness points of related work. Section 4 describes the components and exemplary operation of a smart living room. Section 5 overviews our solution and provides details of the proposed architecture. Section 6 presents a synthesis of the proposed architecture. Section 7 concludes the paper.

2. Related Work

The software architecture of a program or computing system is the structure of the system that comprises software components, the externally visible properties of those components, and the relationships among them [2]. Several interesting architectures and middleware systems for smart spaces have been developed in the last years. Several of them were layered architectures most of which have used agent technologies like the ones proposed by [3-18]. The Smart-M3 was based on a blackboard architecture model [19]. Some proposed architectures were pure agents (resp. multi-agent) system(s) like [20-25]. A web service architecture for the home was proposed by [26]. Some of the proposed architecture were pure sensors network architecture like the ones proposed by [27, 28]. D. J. Cook *et al.*, [29] surveyed some smart space multi-agent systems in order to provide an overview of the role of multi-agent research in the context of smart environments. They concluded that many smart environment software agents rely on expert systems or machine learning technology. Once multiple residents enter a smart environment, the services provided by the smart environment become much more difficult to realize; tracking multiple residents is clearly a challenging problem and in fact has been proven to be NP-Hard. According to the authors, researchers who work in this field acknowledge that the problems of scalability are daunting. Another challenge for multi-agent smart environment software architectures is to define lightweight, simple, and scalable methods. According to Nakajima T. [30], one of the ways to reduce development cost is to reuse existing software as much as possible; these abstractions are useful to build ambient intelligences without considering low-level details. He described four middleware infrastructures developed in their project for building ambient intelligence environments. He affirmed that his middleware infrastructures offer high level abstraction for specific application domains to hide complexities, such as distribution and context-awareness. Memon *et al.*, [31] conducted a literature survey of state-of-the-art Ambient Assisted Living (AAL) frameworks, systems, and platforms to identify the essential aspects of AAL systems and investigate the critical issues from the design, technology, quality-of-service, and user experience perspectives. They found that most AAL systems are confined to a limited set of features and ignore many of the essential AAL system aspects. Standards and technologies are used in a limited and isolated manner, while quality attributes are often insufficiently addressed.

Most of the proposed architecture do not deal in depth with context-awareness when designing smart spaces, which is a key enabling feature of such spaces. Even architecture that addressed context-awareness used the term in general manner and do not clearly specify the components of the context, which definition is used, or whether the context is defined clearly and concisely. Some of the proposed architecture do not contain a module for context modeling or context reasoning; this indicates the negligence of context-awareness, which in turn has a great impact on proactive services adaptation. Some proposed architecture are fully hardware platforms that are based on sensor networks, and the software aspect is almost absent. However, building smart spaces that are considered to be complex systems requires that the developer has an abstraction of physical details of context gathering by sensor networks. Abstraction is a powerful tool to deal with complex systems. Several research projects have focused on a specific aspect of a smart space, which limits their deployment. Some of them focus on context modeling and do not present a context-aware services adaptation module. Others focus on the physical architecture of sensor networks or context-aware content adaptation. The overwhelming majority of proposed architecture have focused on context-aware services adaptation without having a module for context management or modeling. Certain specifics of smart spaces are not taken into account, such as high heterogeneity of available devices and dynamics of environment. Some architectures suffer from being scalable and fault tolerant. Others do not take into account some important criteria of software architecture related to smart spaces, such as loose coupling, flexibility, and interoperability. The produced systems are still unreliable, expensive, and difficult to use.

3. Smart Living Room

3.1. Description of an Exemplary Smart Living Room

An exemplary closed smart space, especially for habitation, is composed of a set of equipment that consists of four main types: a) a multimedia system composed of appliances, which can include a smart TV, satellite receiver, home cinema, *etc.*; b) a climate system, which can be composed of a heater, air conditioner, air purifier, *etc.*; c) light system, which can be composed of window blinds, a set of light bulbs, *etc.*; and d) a communication system, which can be a land-line phone or the user's mobile phone. In addition to some furniture like sofa, table, *etc.* All these devices should provide a set of services through different forms (or modes) to the user or users occupying the closed smart space. These services should take into account the user’s preferences and should be triggered (resp. changed) according to the current context collected from different sensors installed in the closed smart space (Figure 1).

3.2. Overall Operation of the Smart Living Room

In the initial state of the living room (empty), everything is off (TV, radio/music player, light, air conditioning, heater, window blinds closed, *etc.*). At the entrance of the user and detection of his presence, the lighting system composed of the window blinds and light bulbs starts first to adjust the light to the preferences of the user and the current context. Then, the climate system composed of an air cooler, heater, and air purifier starts to adjust the room temperature and air condition to the user's preference and according to the current context. The former system will trigger only if the system perceives at least one user seated on the sofa to avoid wasting energy if the user makes a simple entry and exit from the living room without the intention of staying. After a small period of time (δT), the multimedia system then starts to turn on the TV or radio/music player and home cinema depending on the preferences of the user if he is alone or surrounded. Finally, after another small period of time (ΦT), the communication system starts to provide services tailored to the user's context, such as forwarding the ring of an incoming call to the home cinema if the noise is too high in the living room or display a picture of a phone on the TV screen in order to attract the user's attention.

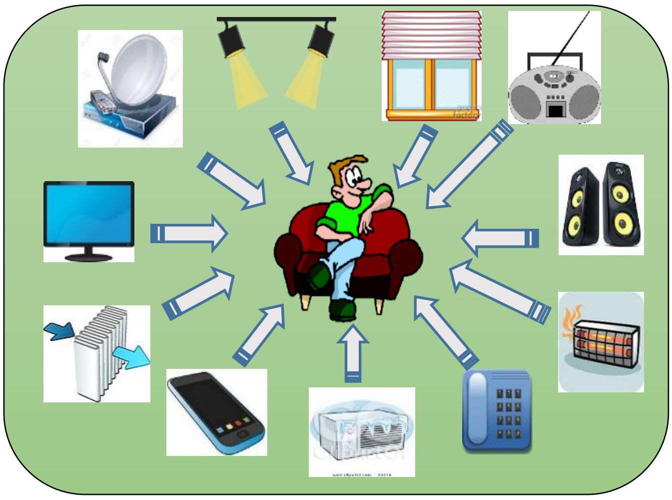

Figure 1. Components of an Exemplary Smart Living Room

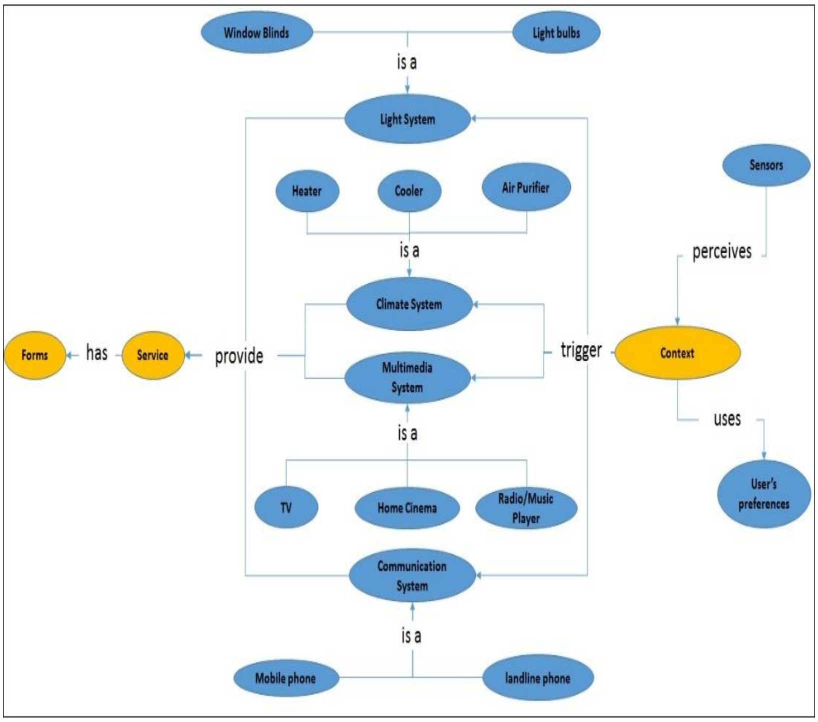

Figure 2. Ontology Based Model of the Smart Living Room

The dynamic aspect of the smart living room described in the above scenario can be modeled using a simple timed automaton (Figure 3) to show the state change over time. There are two main states of the smart living room: empty and occupied. In the first state (*i.e.*, initial and final state), the network sensors perceive that there is no one in the living room, so all the equipment should be off. As soon as the network sensors perceives the presence of one or more user, the smart living room moves to the second state, which in turn contains four sub-states that describe the preferred order of system booting. The light system will start first (occupied 0), and after perceiving at least one user seated on the sofa, the climate system (occupied 1) will start. Then, after a small period of time, the multimedia system will start, and finally after another small period of time the communication system will start. At any time during the occupied state of the smart living room, if the network sensors perceive that there is no body in the room (presence=0), the smart living room will transit directly to the empty state.

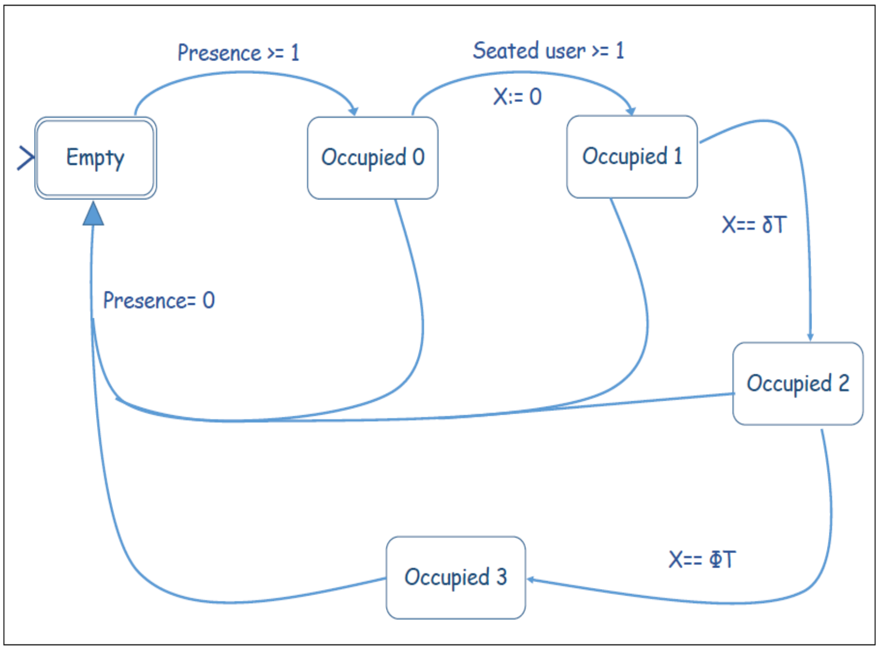

Figure 3. Modeling the Dynamic Aspect of a Smart Living Room

4. Multi-Agent Architecture for a Smart Living Room

In this section, we present the proposed multi-agent architecture that should satisfy the requirements of a smart space in general and a smart living room in particular.

The software architecture of a program or computing system is the structure of the system that comprises of software components, the externally visible properties of those components, and the relationships among them [32]. The main aim of our work is to propose a general purpose software architecture to support the development of closed smart spaces. The architecture should address some important design issues of smart space applications in order to overcome weaknesses of previous architectures. The first objective is to avoid a domain-specific architecture in order to increase its usage.

The second objective is to propose an architecture with a high degree of modularity; this has a positive impact on several software qualities, such as modifiability, reusability, maintainability, extensibility, *etc*. Multi-agent systems have gained an important value in software engineering as solutions for highly distributed problems in dynamic environments. Multi-agent systems offer a decentralized architecture while keeping the autonomy and proactivity of agents. Such characteristics enhance the architecture modularity and fit requirements of appliances and equipment in smart spaces to provide adapted services to inhabitants in a proactive manner according to the current context. Multi-agent systems are comprised of components called agents that are able to interact, usually by passing messages.

The architecture is composed of three main modules: the sensor multi-agent system, the core agent, and the actuator multi-agent system. The overall view of the proposed architecture is shown in Figure 4. In the following, we will detail each module and give particular focus to the core agent, which contains the context modeling and reasoning submodule in order to show the context-awareness aspect of the architecture.

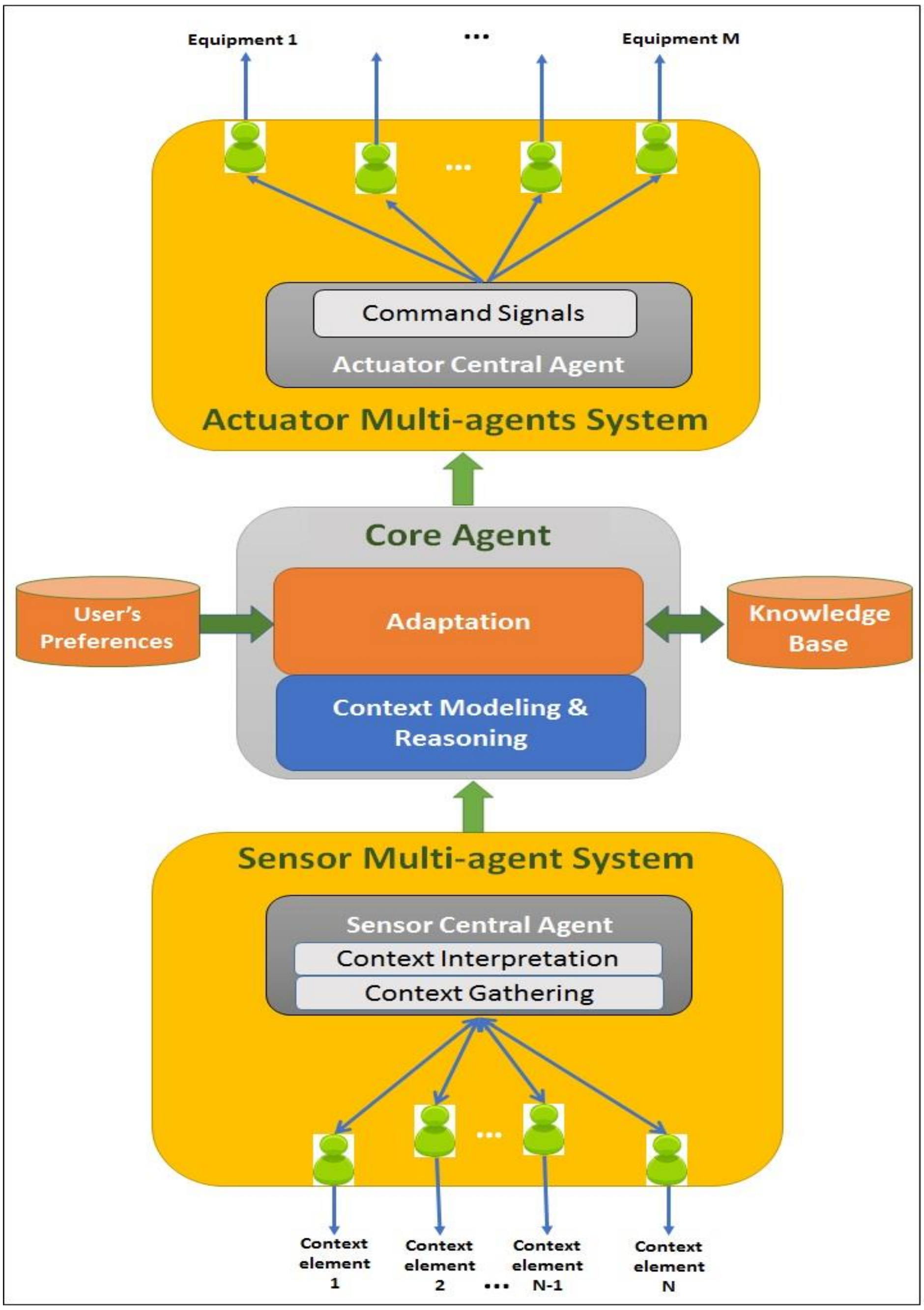

Figure 4. Multi-Agent Architecture for the Smart Living Room

4.1 Sensor Multi-Agent System

The sensor multi-agent system is considered as the physical layer of the architecture. It has the responsibility of gathering raw contextual information using a sensor network.

In addition, it assures interpretation of raw context in order to be useful by the core agent. This multi-agent system is organized as a client/server model where the sensor central agent plays the role of server, and the sensors play the role of clients. Each bit of context information is gathered using a specific sensor, and an agent is assigned to each sensor. The main role of this agent is to sense the value of specific context information—either for each period of time or on demand from the sensor central agent—and send the gathered raw value to the server. The sensor central agent embeds two modules: a) context gathering, which re-assembles all received context data from sensor agents and builds the global current context vector; and b) context interpretation, which has the task of interpreting each raw context data into useful information. For example, a sensed temperature value of -20 degrees Celsius could be interpreted as freezing temperature or a sensed time of value 10 a.m. could be interpreted as morning. The sensor central agent should send the global interpreted current context to the core agent each time it detects one or more changes in the values of the current context. To this end, the sensor central agent uses a vector to store the current context that is updated each time when the global context undergoes a change. This vector is also used to know whether the global context has changed or by comparing the received context values from sensor agents to the content of this vector. If a change is perceived, then this vector is updated and sent to the core agent; otherwise, no action is to be done. Adding or removing sensor agents to perceive context can be done easily according to the requirements of the application without affecting the overall operation of the multi-agent system.

4.2. Core Agent

The core agent is the main module of the architecture in the sense that it contains the two submodules that assure context-awareness of the architecture: context modeling and reasoning and context-aware services adaptation.

4.2.1. Context Modeling and Reasoning: Context-aware applications should provide proactively (without explicit user intervention) adapted services to both users and applications according to the current context. Context awareness is a highly desirable property for smart spaces. Context is considered as the key enabling feature in context-aware applications, and it should be well understood and represented in a clear and concise manner. Before beginning the context modeling phase, it is essential to define context and establish its components. In the literature, several definitions of context were proposed, and some of them were based on enumerating contextual information (localization, nearby people, time, date, *etc.*) like those proposed by [33-35]. Others were based on providing more formal definitions in order to abstract the term, like the one proposed by Dey [36]. However, most of these definitions were specific to a particular domain, such as human-computer interaction and localization systems. In our previous work [37-38], we made a survey of existing definitions of context and proposed a service-oriented definition of context for pervasive and ubiquitous computing environments as follows: "Any information that triggers a service or changes the quality (form or mode) of a service if its value changes." This definition is sufficiently abstract and helps to limit the set of contextual information. We believe that this definition is more expressive, because it is simple, clear, and complete; in addition, it covers all aspects of context. The first step in establishing context elements of the smart living room consists of specifying for each equipment the provided service and the set of information that could trigger the service (Table 1). The second step consists of specifying for each service the set of forms through which the services can be provided. We should also specify for each form of service the set of information whose change will change the form of a service (Table 2). The last step consists of making the union of the two previous sets to get the final list of contextual information and define the set of possible values for each context element (Table 3). This information will compose the global context.

Table 1. Service Triggering Information

Equipment	Provided service	Triggering information
Window blinds	Lighting	User's presence
TV & satellite receiver	Entertainment	Seated users
Home cinema	Sound	Seated users
Mobile & landline phone	Communication	Incoming call
Heater	Heating	Seated users
Cooler	Cooling	Seated users
Air purifier	Air conditioning	Seated users
Radio & music player	Entertainment	Seated users
Light bulbs	Lighting	User's presence

Table 2. Service's Form Changing Information

Equipment	Service's forms					Form changing information
Window blinds	Closed	Mostly closed	Half opened	Mostly opened	Totally opened	User's presence, indoor and outdoor light, user's preferences
TV & satellite receiver	Off	On preferred channel	On other channel			Day type, time, surrounding, user's preferences
Home cinema	Off	Low	Average	High	On preferred	Day type, time, surrounding, user's preferences
Mobile & landline phone	Default mode	Divert to answer machine	Divert to home cinema	Divert to home cinema & icon on TV		Noise, surrounding, time, day type, light
Heater	Off	On preferred	On other			Outdoor temperature, user's preferences
Cooler	Off	On preferred	On other			Outdoor temperature, user's preferences
Air purifier	Off	On low	On average	On high		Air quality, time, noise
Radio & music player	Off	On preferred radio station	On preferred song	On other		Day type, time, surrounding, user's preferences
Light bulbs	Off	On low	On average	On high		User's preferences, indoor & outdoor light, user's preferences, time

Table 3. Context Values

Context	Values				
Seated users	0	1	several		
User's presence	Yes	No			
Indoor light	Dark	Low	Average	High	
Outdoor light	Dark	Low	Average	High	
Noise	Silent	Low	Average	High	
Time	Morning	Afternoon	Evening	Night	Late night
Day type	Work day	Holyday			
Surrounding	Alone	surrounded			
Temperature	Very low	Low	Average	High	Very high
Incoming call	Yes	No			
User's preferences	To be fixed according to the user				

The aim of context modeling is to provide an abstraction of context information from the technical details of context sensing that allows an easy context management and more flexible sharing among devices and appliances. Several approaches were proposed for the modeling and representation of context. A survey made by Strang *et al.* [39] contains an interesting comparative study of different modeling methods. They concluded that ontology makes the best description of context compared to the surveyed methods and seems to be the most suitable tool for context modeling because it provides a good sharing of information with common semantics. A detailed discussion of these methods is out of the scope of this paper, but the authors distinguished the following models for context representation: a) Key-value models, b) Mark-up scheme models, c) Graphical models, d) Object-oriented models, e) Logic-based models, and f) Ontology-based models. Ontology is a set of structured concepts that are organized in a graph where relations can be either semantic relations or composition and heritage relations. The main objective of ontology is to model the set of knowledge of a given domain. It provides a representative vocabulary for a given domain and permits a consistent interpretation of data [40]. Ontology is a powerful tool for knowledge sharing, reuse, and expression of complex situations compared to other data modeling techniques. In addition, ontology supports formal logical reasoning.

In our previous work [41], we have proposed an ontology-based context model for a smart living room that can be easily adapted and used for any closed smart space, as shown in Figure 5. The context ontology model is built based on our context definition where the main class is context and the information that trigger services or change the form of services when their values change are concepts related with the main class in a specialization relationship. Among the advantages of using ontology-based context modeling is the possibility to carry out logical reasoning. It permits the system to deduce relevant new contextual information that cannot be explicitly provided by sensors embedded in the living room. Ontology reasoning permits the system to achieve three main tasks: a) check context inconsistency and conflicts caused by imperfect sensing of contextual information, b) check whether concepts are non-contradictory, and c) find subsumption relationships between classes and instances (for instance, Sunday is a holy day because the weekend is a part of a holiday). There are many automated tools for ontology reasoning; in our work, we used the pellet ontology reasoner [42] to check the consistency of our context ontology and infer new implicit context information.

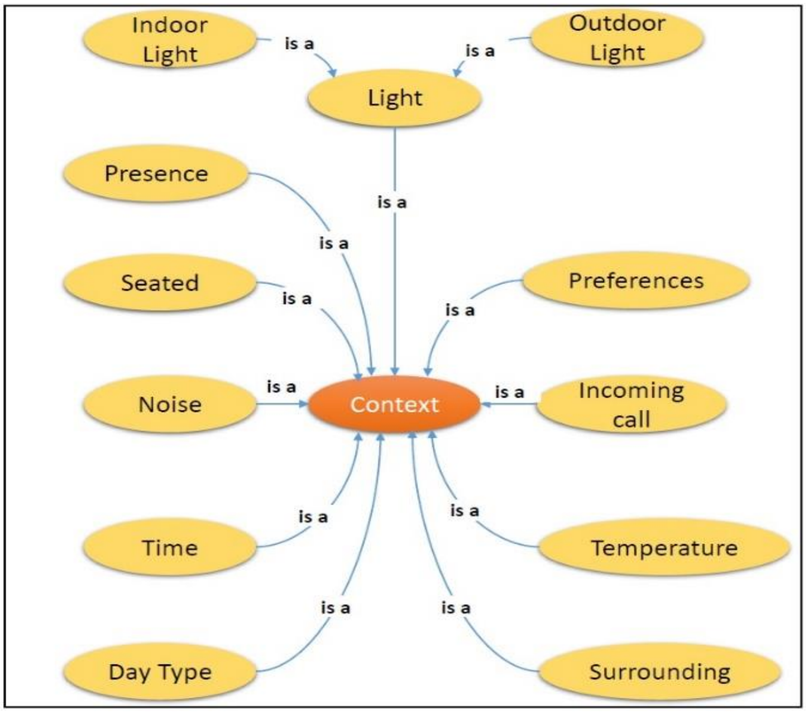

Figure 5. Ontology-Based Context Model for the Smart Living Room

4.2.2. Context-Aware Services Adaptation: The main role of this module is to provide a context-aware services adaptation based on the current context and taking into consideration the user's preferences. The current context values are retrieved from the context modeling and reasoning module. The user's preferences are stored in a database, which should be filled by the user before smart living room operation. For each equipment, the user should provide the preferred form of service according to some context values. For example, the user could prefer to watch the news channel on his TV from 8 PM to 10 PM when he is alone. This module should use an approach of machine learning in order to achieve the adaptation task. This approach can be done either by learning a user's habits or by learning from a set of training examples. Both learning methods need a knowledge base to learn how to map current context and a user's preferences to the appropriate services configuration. In the case of using a case-based approach, the knowledge base will play the role of a base case. The output of this module is a service configuration vector containing for each equipment the appropriate services form. One possible service configuration is shown in Table 4.

Table 4. Example of a Service Configuration Vector

Window blinds	Mostly closed
TV & satellite receiver	On preferred channel
Home cinema	Low
Mobile & landline phone	Divert to home cinema
Heater	Off
Cooler	On preferred
Air purifier	On low
Radio & music player	Off
Light bulbs	Average light

4.3. Actuator Multi-Agent System

This multi-agent system is composed of an actuator central agent (server) and a set of agents (clients) where each equipment of the smart living room is assigned to an agent (*e.g.*, TV agent, heater agent, and window blinds agent). The actuator central agent contains a unique module called command signals that receives the service configuration vector from the adaptation module of the core agent and sends command signals to each equipment agent. In order to minimize the communication stream, the actuator central agent holds a cache vector containing the last services configuration. This cache vector contains the current state of the equipment operation in the smart living room. The central actuator agent sends command signals only to equipment agents that have to change their form of services. The equipment agents have the responsibility to translate the interpreted value to a quantitative value. For example, if the light bulb agent receives a command signal with a value that represents low light, it should translate it to the quantitative value of 10 watts.

4.4. Hardware Infrastructure of the Smart Living Room

The smart living room is based on a hardware infrastructure (Figure 6) that is composed of the following elements:

- A set of sensor nodes to perceive the context elements' values where some of them are specialized to sense one particular context type (*e.g.*, a temperature sensor), and others are grouped together in order to sense one context (*e.g.*, a user's presence that can be sensed by combining a motion sensor and entry/exit sensor). Some sensors are logical, for example, the time and day type sensors, which can be perceived using a simple software calendar and clock. All physical sensors have the ability to wirelessly communicate to sensed context values.
- A sensor gateway sensor, which plays the role of coordinator between all physical sensor nodes. It receives the sensed values of context using a wireless connection with the set of physical sensors. It communicates the vector of sensed context to a dedicated computer using a cabled USB connection.
- The IR transceiver is a USB connected IR (Infra-Red) receiver and transmitter, which has the role to control the set of equipment of the smart living room. Knowing that all the equipment are controlled using a remote control in addition to the manual control, the IR transceiver has the ability to learn IR code of each equipment in order to operate as a universal remote control.
- A dedicated computer that contains the smart living room application; its role is to keep the application running.
- The set of equipment of the smart living room, as mentioned before.

5. Synthesis

The proposed architecture enables developers to build a real context-aware application for smart closed spaces in general and smart living rooms in particular. One of its major components is the context modeling and reasoning module, which assures ontology-based context modeling. Such a modeling approach is a powerful tool for knowledge sharing, reuse, and expression of complex situations compared to other data modeling techniques. In addition, it supports formal logical reasoning, which permits the system to deduce new contextual information that cannot be explicitly provided by sensors embedded in the living room. The context elements are generated based on a clear and concise definition of context that helps to limit the set of context information and covers all aspects of context.

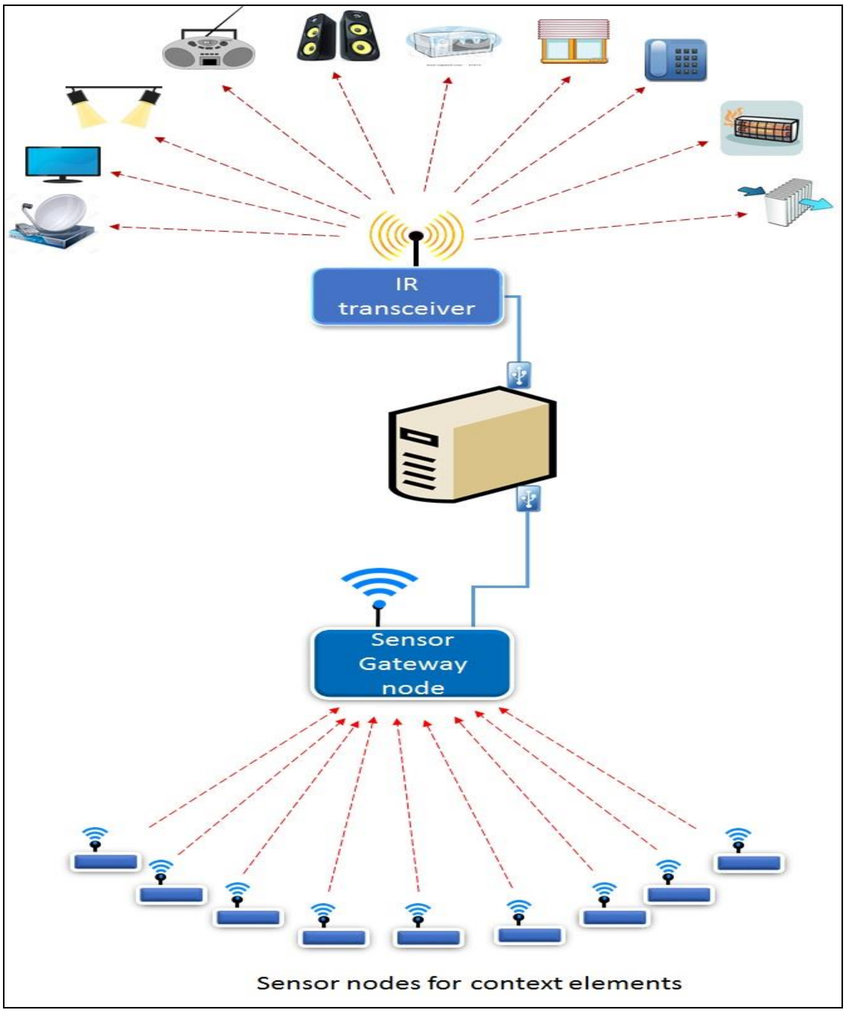

Figure 6. Hardware Infrastructure of the Smart Living Room

The used approach to generate context elements is based on clear steps. The architecture has a high level of modularity and loosely coupled modules with high cohesion, which enhances their reusability in other applications and ease their modifiability (maintenance). The tasks of the smart living room are divided into three main modules: a) sensing current context assured by the multi-agent sensor system, b) context modeling/reasoning and context-aware services adaptation assured by the core agent, and c) command of equipment of the smart living room assured by the actuator multi-agent system. The interaction between the three main modules of the architecture is minimal in the sense that the core agent module receives a vector of current context from the sensor multi-agent system and sends a service configuration vector to the actuator multi-agent system; there is no direct interaction between the sensor multi-agent system and the actuator multi-agent system. Each of the three modules can be developed separately and independently from other modules, which improves the development time of applications using this architecture.

New equipment can be easily added to the smart living room without affecting the whole operation of the system. This is accomplished by simply adding a new agent for that equipment in the actuator multi-agent system and by establishing and adding the set of context information that trigger the service, and finally add sensor agents to the sensor multi-agent system. Such simplicity of adding new equipment considerably enhances the scalability and extensibility of the application.

We have proposed a system architecture for a smart living room rather than a pure software or hardware architecture. The context-awareness of the system such as context modeling/reasoning and services adaptation are assured by the software part of the architecture. The context sensing is assured by the set of physical sensors, and the equipment command is assured by the IR transceiver.

6. Conclusion

The field of smart spaces has attracted considerable attention in the past few years due to the ever increasing availability of cheap sensors, actuators, and the appearance of home appliances with computing power. Most proposed architectures that support the development of such spaces suffer from either not being context-aware or being hardware-oriented, which limits their use. In this paper, we presented a context-aware multi-agent architecture for a smart living room. The proposed architecture focuses on the context-awareness aspect, which is a key enabler for service adaptation in smart spaces. We proposed an ontology-based context modeling approach based on a clear and concise definition of context, which helps limit the set of context information in a few clear steps. The proposed architecture does not consist only of software but also of hardware. It has a high level of modularity with low coupling and high cohesion inside modules, which improves its reusability and maintainability. In addition, it has good extensibility and scalability. Currently, we are implementing the three modules of the architecture in addition to the hardware infrastructure of the smart living room; then, we will proceed to achieve the necessary tests and evaluation of the implementation.

Acknowledgments

This work is supported by the King Abdulaziz City for Science and Technology (KACST), KSA under Grant AC- 35-95. We would like to thank them for giving us the opportunity to perform this work.

References

[1] D. Cook and S. Das, “Smart environments: Technology, protocols and applications”, John Wiley & Sons. ISBN: 0-471-54448-5, **(2004).**

[2] L. Bass, P. Clements and R. Kazman, “Software architecture in practice (2nd ed.)”, Read¬ing, MA: Addison-Wesley, **(2003)**

[3] D. J. Cook, M. Youngblood, E. O. Heierman, K. Gopalratnam, S. Rao, A. Litvin and F. Khawaja, “MavHome: An Agent-Based Smart Home PERCOM '03”, Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, **(2003),** pp. 521-524

[4] D. Zhang, T. Gu and X. Wang, “Enabling Context-aware Smart Home with Semantic Web Technologies International Journal of Human-friendly Welfare Robotic Systems, **(2005)** pp. 12-20

[5] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura and E. Jansen, “The Gator Tech Smart House: a programmable pervasive space”, IEEE Computer Society, vol. 38, no. 6, **(2005)**, pp. 50-60

[6] D. Lopez-de-Ipina, A. Almeida, U. Aguilera, I. Larizgoitia, X. Laiseca, P. Orduna, A. Barbier and J.I. Vazquez, “Dynamic discovery and semantic reasoning for next generation intelligent environments Intelligent Environments”, IET 4th International Conference on, Seattle, WA, **(2008)**, pp.1-10

[7] N. Dimakis, J. K. Soldatos, L. Polymenakos, P. Fleury, J. Curin, and J. H. Kleindienst, “Integrated Development of Context-Aware Applications in Smart Spaces”, Pervasive Computing, IEEE, vol.7, no.4, **(2008)**, pp. 71-79

[8] I. Marsa-Maestre, M. A. Lopez-Carmona and J. R. Velasco, “A hierarchical, agent-based service oriented architecture for smart environments”, Service Oriented Computing and Applications, , springer, vol. 2, no. 4, **(2008)**, pp 167-185

[9] Y. JIN, R. WANG, H. HUANG and L. SUN, “Agent-Oriented Architecture for Ubiquitous Computing in Smart Hyperspace”, Wireless Sensor Network, Scientific Research, vol. 2, no. 1, **(2010)**, pp. 74-84.

[10] G. Lehmann, A. Rieger, M. Blumendorf and S. Albayrak, “A 3-Layer Architecture for Smart Environment Models A model-based approach”, 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), Mannheim, **(2010)**, pp.636 – 641.

[11] M. Aiello, F. Aloise, R. Baldoni, F. Cincotti, C. Guger, A. Lazovik, M. Mecella, P. Pucci, J. Rinsma, G. Santucci and M. Taglieri, “Smart Homes to Improve the Quality of Life for All”, 33rd Annual International Conference of the IEEE EMBS Boston, Massachusetts USA, **(2011)**, pp.1777-1780

[12] V. Degeler, L. I. L. Gonzalez, M. Leva, P. Shrubsole, S. Bonomi, O. Amft and A. Lazovik, “Service-Oriented Architecture for Smart Environments”, Service-Oriented Computing and Applications (SOCA), IEEE 6th International Conference on, Koloa, HI, **(2013)**, pp. 99-104

[13] W. Xiao-chi, X. Jie and F. Zhi-gang, “Design of Smart Space Context-aware System Framework”, Information Technology Journal, vol. 12, no. 20, **(2013)**, pp. 5616-5620.

[14] V. Degeler, L. Gonzalez, M. Leva, P. Shrubsole, S. Bonomi, O. Amft and A. Lazovik, “Service-Oriented Architecture for Smart Environments”, In Proceedings of the IEEE International Conference on Service Oriented Computing and Applications, Koloa, HI, **(2013)**, pp. 99 - 104

[15] D. J. Cook, A. S. Crandall, B. L. Thomas and N. C. Krishnan, "CASAS: A Smart Home in a Box", Computer, vol.46, no. 7, **(2013)**, pp. 62-69.

[16] V. Degeler and A. Lazovik, “Architecture Pattern for Context-Aware Smart Environments”, Chapter 5, Creating Personal, Social, and Urban Awareness through Pervasive Computing, IGI global publishing, **(2014)**, pp. 108-130.

[17] G. Pan, L. Zhang, Z. Wu, S. Li, L. Yang, M. Lin and Y. Shi, “Pervasive Service Bus: Smart SOA Infrastructure for Ambient Intelligence”, IEEE Intelligent Systems, IEEE Computer Society, vol. 29, no. 4, **(2014)**, pp. 52 – 60.

[18] J. Madhusudanan, S. Hariharan, A. S. Manian and V. P. Venkatesan, “A Generic Middleware Model for Smart Home”, International Journal of Computer Network and Information Security, vol. 6, no. 8, **(2014)**, pp. 19-25

[19] J. Honkola, H. Laine, R. Brown and O. Tyrkkö, “Smart-M3 Information Sharing Platform, Computers and Communications (ISCC)”, IEEE Symposium on, Riccione, Italy, **(2010)**, pp. 1041 - 1046

[20] H. Chen, T. Finin and A. Joshi, “A Context Broker for Building Smart Meeting Rooms”, Proceedings of the AAAI Symposium on Knowledge Representation and Ontology for Autonomous Systems Symposium, **(2004)**, pp. 53-60

[21] H. Hagras, V. Callaghan, M. Colley, G. Clarke, A. Pounds-Cornish and H. Duman, “Creating an Ambient-Intelligence Environment Using Embedded Agents”, IEEE inteligent systems magazine, vol. 19, no. 6, **(2004)**, pp.12-19

[22] W. Jih, J. Yung-jen Hsu, T. Lee and L. Chen, “A Multi-agent Context-aware Service Platform in a Smart Space Journal of Computers”, vol. 18, no. 1, **(2007)**, pp. 45-60.

[23] J. C. Augusto and J. O’Donoghue, “CONTEXT-AWARE AGENTS The 6Ws Architecture”, In Proceedings of the International Conference on Agents and Artificial Intelligence, Porto, Portugal, **(2009)**, pp. 591-594

[24] D. I. Tapia and J. M. Corchado, “Ambient Intelligence Based Architecture for Automated Dynamic Environments”, International Journal of Ambient Computing and Intelligence, vol. 1, no. 1, **(2009)**, pp. 15-26.

[25] C. Reinisch, M. J. Kofler and W. Kastner, “ThinkHome: A Smart Home as Digital Ecosystem”, 4th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST), Dubai, UAE, **(2010)**, pp. 256-261.

[26] M. Aiello and S. Dustdar, “Are our homes ready for services? A domotic infrastructure based on the Web service stack, Pervasive and Mobile Computing, vol. 4, no. 4, **(2008)**, pp. 506–525.

[27] M. S. Familiar, J. F. Martınez and L. Lopez, “Pervasive Smart Spaces and Environments: A Service-OrientedMiddleware Architecture for Wireless Ad Hoc and Sensor Networks”, International Journal of Distributed Sensor Networks, Hindawi Publishing Corporation, **(2012)**, pp.1-11

[28] Q. Sun, W. Yu, N. Kochurov, Q. Hao and F. Hu, “A Multi-Agent-Based Intelligent Sensor and Actuator Network Design for Smart House and Home Automation”, Journal of Sensor and Actuator Networks, vol. 2, **(2013)**, pp. 557-588

[29] D. J. Cook, “Multi-agent smart environments”, Journal of Ambient Intelligence and Smart Environments IOS Press, vol. 1, **(2009)**, pp. 47–51

[30] T. Nakajima, “Case Study of Middleware Infrastructure for Ambient Intelligence Environments Chapter Handbook of Ambient Intelligence and Smart Environments”, Springer-Verlag, **(2010)**, pp 229-256

[31] M. Memon, S. R. Wagner, C. F. Pedersen, F. H. A. Beevi and F. O. Hansen, “Ambient Assisted Living Healthcare Frameworks”, Platforms, Standards, and Quality Attributes. Sensors, vol. 14, **(2014)**, pp. 4312-4341.

[32] L. Bass, P. Clements and R. Kazman, “Software architecture in practice”, (2nd ed.). Read-ing, MA: Addison-Wesley, **(2003).**

[33] S. Schilit and M. Theimer, "Disseminating Active Map Information to Mobile Hosts", IEEE Nenvork, vol. 8, no. 5, **(1994)**, pp. 22-32.

[34] P.J. Brown, J.D. Bovey and X. Chen, "Context-aware Applications: From the Aboratory to the Marketplace" IEEE Personal Communications, vol. 4, no. 5, **(1997)**, pp. 58-64.

[35] N. Ryan., J. Pascoe and D. Morse, "Enhanced Reality Fieldwork:the Context -Aware Archeological Assistant" Computer Applications in Archeology, **(1997)**
[36] A. K. Dey "Understanding and Using Context" Journal of Personal and ubiquitous computing, vol. 5, **(2001)**, pp. 4-7.
[37] M. Miraoui and C. Tadj, “A service Oriented Definition of Context for Pervasive Computing, in Proceedings of the 16th International Conference on Computing”, Mexico city, Mexico, **(2007)**, pp. 1-6.
[38] M. Miraoui, C. Tadj and C.B. Amar, “Context Modeling and ContextAware Service Adaptation for Pervasive Computing Systems”, International Journal of Computer and Information Science and Engineering, vol. 2, no. 3, **(2008)**, pp. 148-157.
[39] T. Strang and C. Linnhoff-Popien, “A Context Modeling survey”, In the first International Workshop on Advanced context modeling, Reasoning and management, UbiComp, **(2004)**
[40] M. Miraoui, S. El-etriby, C. Tadj and A. Z. Abid, “Ontology-Based Context Modeling for a Smart Living Room”, Proceedings of the World Congress on Engineering and Computer Science,Vol I WCECS 2015, San Francisco, USA, **(2015)**, pp. 127-132
[41] M. Miraoui, C. Tadj and C. B. Amar, “Context modeling and contextaware service adaptation for pervasive computing systems”, International Journal of Computer and Information Science and Engineering (llCISE), vol. 2, no. 3, **(2008)**, pp. 148-157.
[42] E. Sirin, B. Parsia, B. Grau, A. Kalyanpur and Y. Katz, “Pellet: A practical OWL-DL reasoned”, Web Semantics: Science, Services and Agents on the World Wide Web, vol. 5, no. 2, **(2007)**, pp. 51–53.