A Contextual Approach to Invoke Intelligent House Services: An Application to Help Physically Handicapped Persons

Valérie Monfort[1,2] and Fayssal Felhi[3,4]

[1]University of Sfax, ISIMS, Miracle, M'harza Road, Km 1,5 BP 1030, Sfax, Tunisia
[2]University of Paris 1 Panthéon Sorbonne
90 Tolbiac Road, 75634 Paris cedex 13, Paris, France
[3]University of Sousse, ISITC, GP1 Hammam Sousse 4011, Sousse, Tunisia
[4]University of Kairouan, ISIG, Khemais Elouini Road, Kairouan, Tunisia

Abstract. Intelligent Houses are considered as a subclass of Programmable houses that are automatically planed to perform some activities and then, to propose services to inhabitants as persons with specific needs. In this paper, we propose an infrastructure based on geo-localization, Web services and aspects to promote Adaptive Intelligent Houses. Our research work uses a case study extracted from a concrete industrial project.

1 Introduction

Intelligent Houses are very similar to Programmable houses that are programmed to perform some activities. The difference is in a way the house is being programmed. In case of Programmable House technologies, human intervention is needed to program or reprogram the house. In case of Intelligent Houses it would be done automatically by the house itself. This is achieved by the capability of the house to observe the inhabitants in the search for patterns. After the pattern is found the house would start performing the activities automatically every time the pattern would be noticed again. By performing some activities for people, the house can do some important activities like opening the door or switching the light. There is a differentiation in Intelligent Houses between the system's reaction on simple sensor inputs and system recognizing and assessing situations. Some Intelligent Houses own ambient intelligence allowing them to assess situations and scenarios and to be present all the time focusing on its improvements. So eventually the right reliability would be achieved. Other advantages of the Intelligent House are capabilities of intrusion or accident detection. As the house is observing the inhabitant all the time and all their usual activities are stored, then it is possible to easily detect an unusual activity. In some circumstances it could be a sign that a person for example fainted. In such cases the neighbors or health care can be informed. Intelligent houses require specific software to implement high flexibility, reliability and availability level. Such systems usually are very complex and maintenance is hard and costly. Moreover, used devices and

Monfort V. and Felhi F.
A Contextual Approach To Invoke Intelligent House Services: An Application to Help Physically Handicapped Persons.
DOI: 10.5220/0003025000820091
In *Proceedings of the International Joint Workshop on Technologies for Context-Aware Business Process Management, Advanced Enterprise Architecture and Repositories and Recent Trends in SOA Based Information Systems (ICEIS 2010)*, page
ISBN: 978-989-8425-09-6

technologies are very heterogeneous and it is not obvious to add a new device or to develop a new service to satisfy the user.

We noticed that Service Oriented Architecture (SOA) offers a great flexibility to Information Systems (IS) because each application owns interfaces masking implementation details. So, applications own interfaces including services and are seen as black boxes independently connected to a middleware as Enterprise Application Integration bus (EAI) with its connectors and adaptors. Moreover, Web services are based on standards and are till now the cheapest and simplest solution to support interoperability between platforms. Based on Web services, Enterprise Services Bus (ESB) [17] is a kind of services Web based EAI and allows loosely coupling with low costs. However, even if Web services offer technical connection means for interoperability, they address neither adaptability nor context aware adaptability.

We are convinced that platforms for adaptability as [23] could be used as a software solution for an Intelligent Houses. Particularly, Wcomp supports Web services and Aspects Weaving for adaptability implementation. This paper is based on a concrete industrial project improving accessibility for persons with specific needs in their usual environment (at home, in their cars, in the underground...). It shows a concrete implementation of context awareness in Intelligent Houses with Web services and aspects based on WComp Platform. We shall process as followed. The second section presents context awareness. The third section presents the case study extracted from the industrial project. The fourth section explains Web services, and WComp platform. The fifth section presents the architecture and development examples. The Sixth section shows related works and finally the last section treats our conclusion.

2 Context Awareness

The emergence of new technologies in particular wireless communications and the increasing use of portable devices (smart phones, personal digital assistants-PDA-, laptops…) have simulated the emergence of a new computing paradigm called: pervasive computing. In fact we have moved from the desktop computing paradigm to the mobile and ubiquitous computing paradigm. Pervasive computing refers to the seamless integration of devices into the users everyday life. “Appliances should disappear into the background to make the user and his tasks the central focus rather than computing devices and technical issues.”. Computing applications now operate in a variety of new settings; for example, embedded in cars or wearable devices. They use information about their context to respond and adapt to changes in the computing environment. They are, in short, increasingly context aware.

They explain how to use context and propose a classification of the features of context-aware applications that combine their different ideas. They consequently define three categories of features that context-aware applications may support as (1) presentation of information and services to the user; (2) automatic execution of a service; and (3) tagging of context to information for later retrieval. In order to describe how an application can be context aware, a lot of works are described in [33] and propose various architectures for context-aware systems.

All these works converge to a general architecture composed of five ordered layers presented in fig. 1. The complete description of every layer is given in [22].

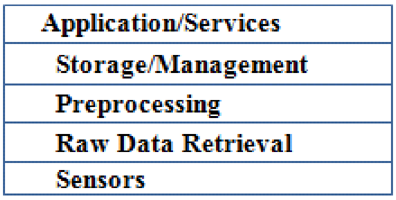

Fig. 1. Architecture of context-aware system.

3 Case Study

3.1 Context

We are working on a concrete Demotic or intelligent house project to help handicapped persons for their usual tasks at home. They can allow continuing an independent life for elderly persons (control, communicate, banking….) and those persons can thus benefit from Domotics to be more autonomous. So, Domotics can contribute to a better quality of life. Moreover, an intelligent (smart) house can be considered as a comprehensive and intelligent aid, adaptable to the functional possibilities of the user and to the desired actions. For instance, the intelligent house can manage: i) safety as burglar alarm, fire alarm, gas and smoke detector, …ii) signaling as video intercom, computer with programs a peripherals, recognition software, speech module, … iii) personal care and housekeeping as bathroom lift, adjustable bed … iv) communication as person-to-person communication, access to various services such as e shopping and e banking, monitoring health conditions, , ... v) automatic switching lightning, remote control of radio/TV/Video/CD/DVD, remote control of front door … The elements that generate signals are called “sensors” (switches, temperature meters, infrared-cells, motion-detectors, wind-detectors, light-cells,…). We called “actuators” the elements that are activated by the system (relays / dimmers for lamps, motors for the garage door, ...).

In this paper, we, off course, reduced the scope of the case study. In our domotics system sensors, actuators, supply and communication are part of a home network. The network is controlled by the coordination-system (fig.2). Sensors intercept signals and send information to the coordination-system. For instance, when the handicapped person goes close to a door, he enters into the perimeter defined to be detected by a geo-localization sensor. So, while door opening, the geo-localization sensors into the room send entrance information to a geo-localization system including data base that performs the person’s position and associates several services specific to the room. The handicapped person may loudly ask for a service, a sensor detects vocal information to send it to the system and the system provides the required service.

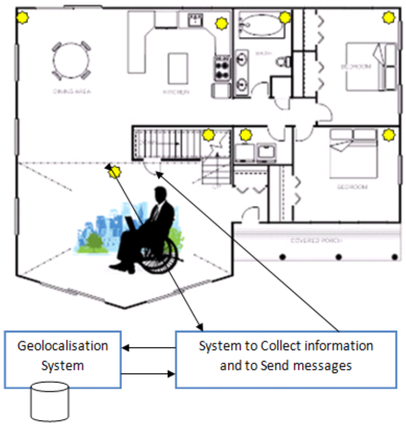

Fig. 2. Intelligent house.

3.2 Modeling

According to previous section, the different services invoked by the handicapped person are: opening the door, switching on the light or TV … The actor (the handicapped person) goes into the room that is equipped by several sensors, and the system “InRoomDeviceManagement” detects that room intrusion (Fig 3.). According to the voice signal the sensor will choose the fitted service. Position, Time, Climate, Voice are signals intercepted by the system to provide services. We aim to offer: flexibility to changes, interoperability to different platforms and adaptability. Let us see now the way we implemented the case study and firstly present the technologies we used as Web services for flexibility and interoperability and Wcomp platform for adaptability.

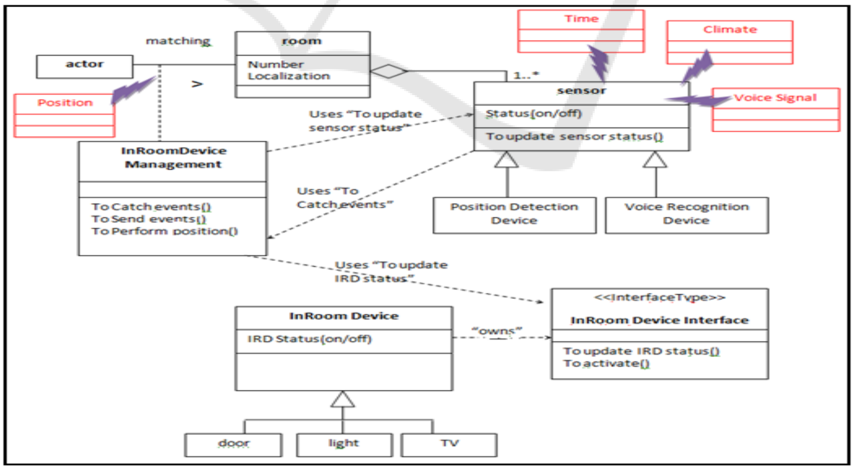

Fig. 3. UML Class model of the case study.

4 Web service, Aspect and WComp

4.1 Web Service Definition

Web services (WS) [7], like any other middleware technologies, aim to provide mechanisms to bridge heterogeneous platforms, allowing data to flow across various programs. The WS technology looks very similar to what most middleware technologies looks like. Consequently, each WS has an Interface Definition Language, namely WSDL (Web Service Description Language), that is responsible for the message payload, itself described with the equally famous protocol SOAP (Object Access Protocol), while data structures are expressed by XML (eXtended Markup Language). Very often, WS are stored in UDDI (Universal Description Discovery and Integration) registry. WS-BPEL (Business Process Execution Language) is a WS orchestration language. An orchestration specifies an executable process that involves message exchanges with other systems, such that the message exchange sequences are controlled by the orchestration designer. WS-BPEL provides a language for the specification of Executable and Abstract business processes. By doing so, it extends the WS interaction model and enables it to support business transactions. WS-BPEL defines an interoperable integration model that facilitates the expansion of automated process integration [12] in both the intra-corporate and the business-to-business spaces. Web services are the fitted solution for flexibility and interoperability but as mentioned in related works [19], [26] they do not offer adaptability. However, in our case study we shall not need UDDI repository locally, inside house. Following section briefly presents Aspects paradigm.

4.2 Aspects Paradigm Definition

Aspect Oriented Programming (AOP) is viewed as an answer to improve Web services flexibility. AOP [1] is a paradigm that enables the modularization of crosscutting concerns into single units called aspects, which are modular units of crosscutting implementation. Aspect-oriented languages are implemented over a set of definitions:

- *Joinpoints*: They denote the locations in the program that are affected by a particular crosscutting concern.
- *Pointcuts*: They specify a collection of conditional joinpoints.
- *Advices*: They are codes that are executed *before*, *after* or *around* a joinpoint.

In AOP, a tool named *weaver* takes the code specified in a traditional (base) programming language, and the additional code specified in an aspect language, and merges together the both in order to generate the final behavior. The weaving can occur at compile time (modifying the compiler), load time (modifying the class loader) or runtime (modifying the interpreter). Let us see now WComp Platform, a platform for context aware adaptation.

4.3 The Wcomp Platform

WComp is a prototyping “development” environment for context-aware applications. The WComp Architecture is organized around Containers and Designers paradigms. The purpose of the Containers is to take into account system services required by Components of an assembly during runtime: instantiation, destruction of software Components and Connectors. The purpose of the Designers allows configuring assemblies of through Containers. To promote adaptation to context WComp uses Aspect Assembly paradigm. Aspect Assemblies can either be selected by a user or fired by a context adaptation process. It uses a weaver that allows adding and or suppressing components. A container includes a set of (Beans) components. Each bean has: properties, input methods that use received input information, and output Methods to send to another bean, for instance, output information. Aspect Assemblies allow defining links between Beans by using input and output information. WComp uses UPnP (Plug and Play) technology to detect locally whether the device is active or not and to define input methods and sent events for each component. With this architecture WComp allows: i) managing devices heterogeneity and dynamic discovering by using UPnP, ii) events driven interactions with devices, iii) managing dynamic devices connection and disconnection in infrastructure. Concerning proposed case study, Beans are composed according to needs in a predefined manner (Lightweight Components Architecture or LCA). Let us see now the proposed solution.

5 Solution

5.1 Architecture

Fig.4 shows proposed architecture where UPnP plug-in intercepts events coming from devices to send them to WComp that performs localization of the person. The Gateway links services to a set of points within a room. This architecture is deployed locally, inside the house of the handicapped person.

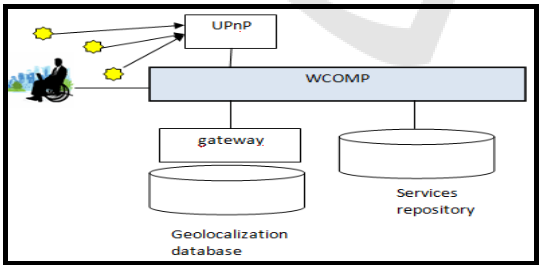

Fig. 4. Proposed Architecture.

5.2 Implementation

Here is a set of figures denoting our application and illustrating the corresponding assembly. Fig.5 shows the request of several services in a sitting room as: Air conditioner, Lightning, Screen Camera Monitoring. When the handicapped person is in the room status is on and the “EventToggler” Bean intercepts this status change and services request is expressed.

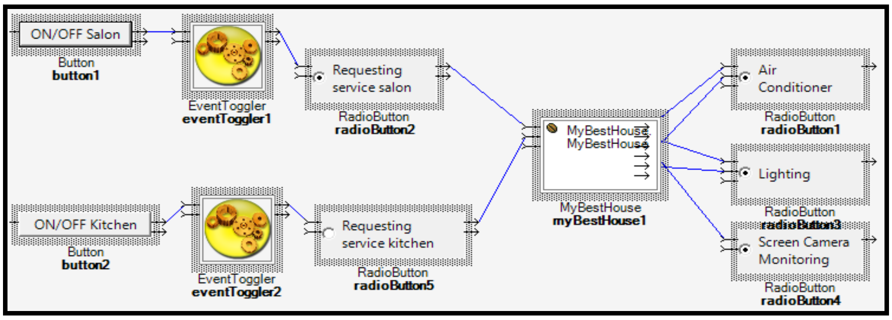

Fig. 5. WComp container modeling.

Following Fig.6 shows the updating status through the WComp interface. We simulated the sensors with this interface (Fig. 6).

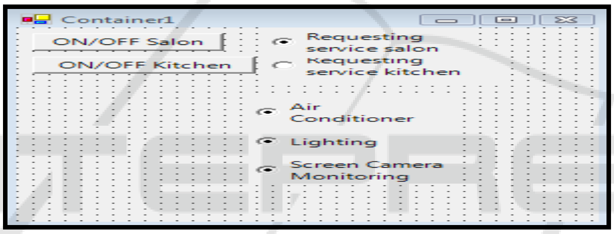

Fig. 6. WComp interface, asking for a service.

Following Fig.7 shows a sequence diagram to express the process. The handicapped person goes into a room and asks for light switch on service. We simulate this with a click “on” and the “EventToggler” Bean intercepts the ok value

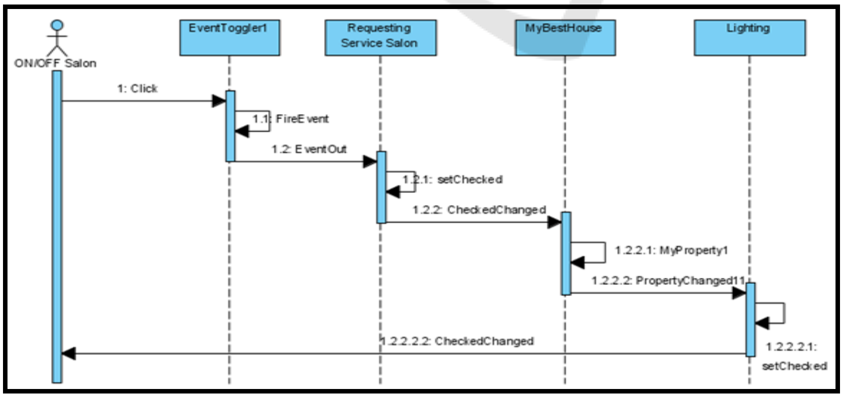

Fig. 7. Sequence Diagram.

(coming from “on” value) and then executes “FireEvent” method that updates the light switch status. This value comes from the event “EventOut” via the bean called « Demande Service Salon » that updates light switch status. « setChecked » changes the status of the radioButon « Demande Service Salon » via the « MyBestHouse » Bean...

Fig. 8 and 9 show the definition of the assembling between “MyBestHouse” Bean and the “radioButton2” in the container (lines 35 and 36) . The link between the ”radioButton2” via “MyBestHouse » defined by “EventHandler” that intercepts “CheckedChanged” value and provides the value to the Entry method to “MyBestHouse” Bean. This value is performed by “this.__radioButton2_to_myBestHouse1_0 » as a parameter of the “EventHandler” method.

```
35 this.radioButton2.CheckedChanged += new
36 System.EventHandler(this.__radioButton2_to_myBestHouse1_0);
37 this.myBestHouse1.PropertyChanged1 += new
38 MyBestHouse.MyBestHouse.IntValueEventHandler(this.__myBestHouse1
39 _to_radioButton1_1);
40 [......]
41}
```

Fig.8. Assembling definition.

The lines 37 to 30 are the assembling between bean “MyBestHouse” Bean and a “radioButton” in the container.

```
42 private void __radioButton2_to_myBestHouse1_0(object sender,
43 System.EventArgs e) {
44 this.myBestHouse1.MyProperty1 = this.radioButton2.Checked;
45 }
46 private void __myBestHouse1_to_radioButton1_1(bool val) {
47 this.radioButton1.Checked = val;
48 }
49 [......]
50 }
```

Fig. 9. Updating radioButton.

The lignes 42 and 43 show the declaration of the method __radioButton2_to_myBestHouse1_0 whose parameter is the object or the component that will receive the sent value. This work allowed us to develop complete application we used concrete sensor and fitted electronics system to define a genuine prototype. This is a part of the future works as shown in following section.

6 Future Works

We would like to extend previous solution by adding the possibility to invoke a distant service via the Web. For instance, the handicapped person wants to command a pizza, but it is Sunday, only a few pizzerias are opened on Sunday. The system analyses the request of the handicapped person adapting it to the specific day and by listing the opened pizzerias on Sunday, close to the house of the user. He just has to choose the pizzeria with any media. WComp can do the adaptation according to the date and the localization. It offers the possibility to invoke a distant service according to a BPEL process. Then it is necessary to develop a correspondence gateway to link service to localization, as the gateway inside house. UDDI repository is required with distant service invocation. We have to work to define gateways and performing geolocalization databases. These geolocalization databases aim to define geographical zones. Moreover, we notice the need to define Aspects Beans and we would like to increase adaptability via auto adaptability, we also are working in this direction.

7 Related Works

Several research works as [25], aim to use meta modeling [10], [13], [14], [20], [21], [22], context awareness platform as WComp [24] or other platform to promote adaptability. In Intelligent House domain more often system used is system based hardware. We did not find any research work using context aware platform.

8 Conclusions

This research paper presents a genuine case study coming from an industrial project about Intelligent Houses for handicapped persons. The context adaptation platform (WComp) is considered as a middleware: orchestrating messages and events sending, reconfiguring components according to aspects, invoking local or distant Web services. We presented an architecture and we pragmatically showed the adaptation with models and pieces of code. This architecture can be extended by offering local and distant services according to different context parameters as position, dates, weather, … We are working now on a full auto adaptable platform.

References

1. Kiczales, G., Lamping, J., Maeda, C., and Lopes, C. (1997). Aspect-oriented programming. In Proceedings European Conference on Object-Oriented Programming, volume 1241, pages 220–242. Springer- Verlag, Berlin, Heidelberg, and New York.
2. V. Monfort, S. Hammoudi, When Parameterized MDD Supports Aspect Based SOA , IJEBR 2010, International Journal of E-Business Research (To appear).
3. V. Monfort, S. Hammoudi, ICSOC, Towards Adaptable SOA: Model Driven Development, Context and Aspect The 7th International Conference on Service Oriented Computing, November 23-27 2009, Stockholm, Sweden.
4. Charfi, A. and Mezini, M. (2004). Aspect-oriented Web service composition with AO4BPEL. In Proceedings of the 2nd European Conference on Web Services (ECOWS), volume 3250 of LNCS, pages 168– 182. Springer.
5. Tidwell, D. (2000). Web services: The web's next revolution
6. S. Vale, S. Hammoudi, Context-aware Model Driven Development by Parameterized Transformation Proceedings of MDISIS 2008
7. Web site retrieved from WSA http://www.w3.org/TR/ws-arch/

8. Chen, G., and Kotz, D. A survey of context-aware mobile computing research. Tech. rep., Dept. of Computer Science, Dartmouth College, 2000.
9. De Farias, C. R. G., Pires, L. F., and van Sinderen, M. A MOF Metamodel for the Development of Context-Aware Mobile Applications. In Proceeding of the 22nd ACM Symposium on Applied Computing (SAC'07) (2007).
10. Dey, A. K. Understanding and Using Context. Personal and Ubiquitous Computing 5, 1 (2001), 4-7.
11. Ferrara, A. (2004). Web services: a process algebra approach. In ICSOC ’04: Proceedings of the 2nd international conference on Service oriented computing, pages 242–251, New York, NY, USA. ACM Press.
12. Frankel S. David., Model Driven Architecture: Applying MDA to Enterprise Computing, Wiley Publishing, Inc(2003)
13. Gottschalk, F. and van der Aalst, Wil M.P. and Jansen-Vullers, M. H. and La Rosa, M. (2008) Configurable Workflow Models. International Journal of Cooperative Information Systems (IJCIS).
14. Gu, T., Pung, H. K., and Zhang, D. Q. A service-oriented middleware for building context-aware services. Journal of Network and Computer Applications, 28 (2005), 1-18.
15. Haddad, S., Moreaux, P., and Rampacek, S. (2006). Client synthesis for web services by way of a timed semantics. In Proceedings of the 8th Int. Conf. on Enterprise Information Systems (ICEIS06), pages 19–26.
16. ESB MULE
17. http://www.mulesource.org/display/COMMUNITY/Home
18. Hmida, M. B., Tomaz, R. F., and Monfort, V. (2006). Applying aop concepts to increase web services flexibility. Journal of Digital Information Management (JDIM), 4(1):37–43.
19. J. Klein, L. Hélouet, and J. M. Jézéquel. -- Semantic-based weaving of scenarios. -- In Proceedings of the 5th International Conference on Aspect-Oriented Software Development (AOSD'06), Bonn, Germany, March 2006. ACM
20. S. Lundesgaard, A. Solberg, J. Oldevik, R. France, J. Oyvind Aagedal, F. Eliassen, Construction and Execution of Adaptable Applications Using an Aspect- Oriented and Model Driven Approach, IFIP DAIS 2007, LNCS 4531, 76-89, 2007
21. J. Y. Tigli, S. Lavirotte, G. Rey, V. Hourdin, D. Cheung-Foo-Wo, E. Callegari, M. Riveill. « WComp Middleware for Ubiquitous Computing: Aspects and Composite Event-based Web Services ». Annals of Telecommunications, volume 64, n° 3-4, pages 197, april 2009. ISSN 0003-4347