Designing a software framework for smart environments

J. I. Sánchez-Venzalá, A. Fernández-Montes, J. A. Ortega, F. J. Cuberos
Department of Computer Science, University of Seville, Seville, Spain
jisanchez@us.es,afdez@us.es,jortega@us.es,fjcuberos@us.es

Abstract

The purpose of this paper is to describe the design of a software framework to build smart environments based in the architecture proposed and the OSGi technology. The Framework divides the general process in three subprocesses, and specifies a methodology for the implementation of each part, as well as the way in which these parts must interact between them. The main objective of this framework is to provide a platform for build smart environments in a modular way, where flexibility and interoperability between modules were guaranteed while implementation is facilitated by reducing the gap between each individual device and the behaviour of the system a a whole, always from the software point of view: protocol, data and interfaces heterogeneity and device interoperability.

1 Introduction

From the beginning of computing, computers have increase their capacity and capabilities, while their size have been reduced. These changes have made possible new applications of computing, like the ambient intelligence, which aims to improve the environment to make life easier for users in a transparent way.

Smart environments are one of the ambient intelligence principal fields, and a representation of the vision of future for society. Thus, recent technological advances have given ambient intelligence a new impulse, and smart environments is now a trending field in the current computer sphere.

In a recent paper we have studied several smart home projects, identifying the principal processes and tasks which have to be carried out in order to implement an smart environment.

The study of the projects, concluded that software layer of smart environments did not receive the appropriate relevance by researchers. For example, interoperability was one of the major problem in smart environment, and software is able to provide solutions for it.

We have proposed a general software architecture taking in consideration these tasks and generalizing them for smart environments.

The general architecture proposed for these tasks is based in the view of smart environments as systems where five main areas [1] interacts between them (Figure 1). These five areas, can be divided in two groups, three main areas which form the automation process of the environment: Sense, Reason and Act, and two support areas: HCI and Security. Each one of these areas is divided for his part, in several tasks, in a hierarchical structure, as is presented in section 3.

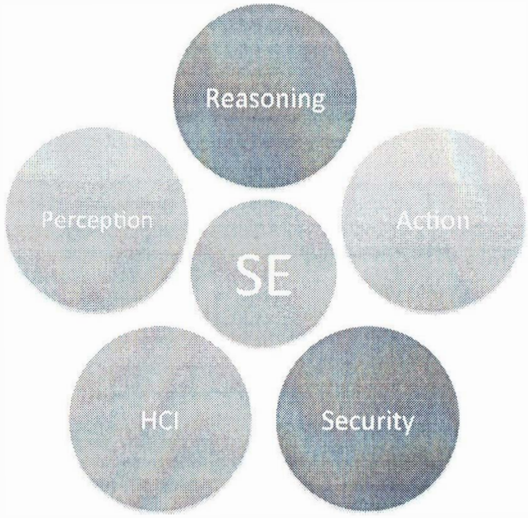

Figure 1: Smart Environment areas.

The framework designed from the architecture philosophy, gives response for the needs of each area, providing a common service oriented middleware (SOA) and defining how each task have to be implemented by following the principles of OSGi (Open Service Gateway Initiative).

The definition of the smart environment software framework provides a general architecture for smart environments, a common service toolkit and gives the software layer the responsibility of devices and services interoperability, reducing the complexity of the smart environment construction process.

2 Related Work

Several proposals are founded in bibliography for architecture and integration in smart environments. In [2], Cook y Das, propose an architecture used in their smart environment project MavHome, which defines a bottom-up design, as the information flows in the smart environment. This architecture divides the model in four layers: physical, communication, information and decision.

Augusto and Nakashima [3], propose a middleware that interacts with each one of the components in the environment: sensors, applications, services, users and the own environment.

In Salvador and Larrea [4], it is affirmed that middleware architectures oriented to services simplify the task of application development for the environment, because they group common functionality for the services and applications in the middleware.

The construction of an adequate architecture, reduce the coupling, and increase the interoperability between devices and software components of the smart environment. The search of an architecture for a framework for smart environments is a research line eminently software, because the problem is an integration problem of the difference software components of the system.

3 Architecture

The architecture is divided as mentioned in introduction in two groups of processes. The first group covers the automation process: perception, reasoning and acting. The second group covers the support tasks in two processes: human computer interaction and security.

3.1 Perception

Perception process objective is to acquire information from the real world which allow to build an accurate representation for it. It has to deal with low level details to retrieve data and incorporate it to a knowledge base or to a real world model (i.e. an ontology).

Perception process is divided in several tasks which covers the whole process as it is shown in Figure 3.1.

- **Data Collector**: retrieves data from physical devices (sensors).
- **Verifier**: verifies data retrieved by collector to avoid outliers.
- **Repairer**: processes data to repair its values if necessary.
- **Filter**: discards unuseful data.
- **Ontologizer**: introduces data into the real world model.

3.2 Reasoning

Reasoning process is a set of services and functions that interacts to achieve the process objective by mean of the study of the data retrieved by perception process. It is divided as shown in Figure 3.2.

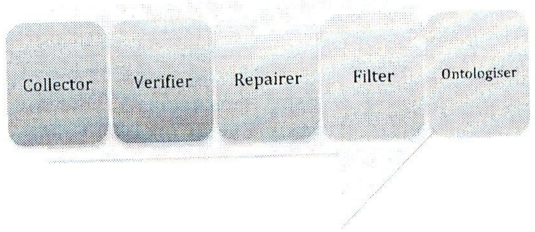

Figure 2: Perception.

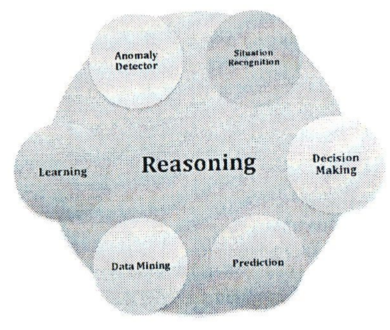

Figure 3: Reasoning.

- **Learning**: adaptation of the environment to the user preferences, needs and habits.
- **Anomaly Detection**: supervise actions and decisions opposites to other user actions.
- **Situation Recognition**: classifies an event in a category.
- **Decision Making**: selects the best action from the different alternatives.
- **Prediction**: make prediction for future actions from the events already occurred.
- **Data Mining**: discover knowledge and patterns in data.

3.3 Acting

Acting process performs actions over the environment to achieve an intelligent behavior. It transforms the decisions taken in the reasoning process in physical actions by mean of actuators. Decisions have to go throughout the difference acting tasks as shown in Figure 3.3.

- **Policy Manager**: implements policies for the actions.
- **Task Scheduler**: prioritize actions for their execution.
- **Task Runner**: receive a set of actions from scheduler and send them to the devices.

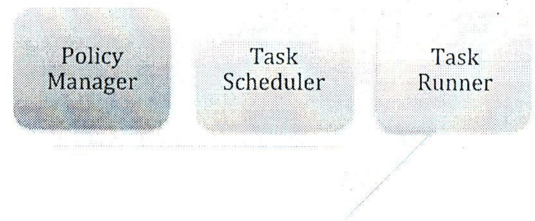

Figure 4: Acting.

4 OSGi

OSGi is a modular framework for Java which defines the way to build software using small, reusable and collaborative modules, called bundles, and the way in which these bundles must interact in runtime. The framework provides for the developer a service oriented environment based in components that use a SOA architecture. The aim is to reduce implementation and integration complexity, favoring interoperability and thus facilitating bundles and services interchange.

OSGi was born as a solution for devices communication (is fully compatible with Jini and UPnP), although it has been applied in many other areas too, like automobile, grid computing, development IDEs (Eclipse), application servers, etc.

The development of the framework using this philosophy, brings dynamic modularity and a standard software integration method.

The framework will be divided in several bundles, each one of them with a determined purpose (defined by the architecture), and an stablished interface.

4.1 Bundles

Bundles are the running unit in OSGi. A Bundle register zero or more services in the service registry which allows the search and location of a service object. This way, a bundle can discover and use services provided by other bundles in a dynamic environment.

Each task defined in the proposed architecture will be implemented by one or more bundles, to divide functionality and force them to satisfy the interface requirements that guarantee the interoperability of the software components.

5 Framework

The framework implements the architecture proposed, like is shown in Figure 5. Where can be observed that the different processes and tasks of the architecture are included.

Bundles exchange information (i.e. messages) through the communication layer, which connects the different software artifacts of the framework.

The framework, which is in an early stage of development, include the two support areas commented in the Introduction, human computer interaction and security, to complete the whole smart environment application functionality.

6 Future Work

This work is the origin for several open paths, which will be followed in the near future. One of these paths is the extension of the architecture, which have to be extended taking into account the support processes. The second path is the implementation of the framework, in order to be evaluated. For this implementation the detailed design of each process and each task is required. Perception is a good starting point for the implementation, because it is the process which generates data for the rest of processes. The third path could be the integration of smart environments solutions, from other authors, in the framework. The framework, thanks to its modularity and low coupling, allows to adapt algorithms to extends framework capabilities and compare between different solutions for a determined task in the same context.

The last objective for the framework is to give a general solution for smart environments construction which allows to integrate easily devices and software artifacts to give response to the requirements.

7 Conclussions

From the work realized for this paper, can be concluded that smart environment projects do not pay attention to the importance of software in the majority of the problems which appear in the construction of smart environments. This study has also detect the lack of an ambient intelligence paradigm which drives the way in which these systems have to be developed. The absence of agreement in this aspect provokes that each project implements its own solution difficulting the comparison between them, although they solve the same problem. A common architecture could define the paradigm to follow and could favor the reutilization of modules between different solutions.

The implementation of the framework, as a practical demonstration of the utility and potential of a common architecture, will establish a research platform for future developments and tests. Verification and evaluation, which are both controversial aspects in ambient intelligence and smart environments because of the complexity of compare in different contexts, will be important tasks which will be possible within this implementation. An added value for the implementation, is the potential use of the development in research projects that could provide with feedback for the evolution of the framework and the architecture.

8 Acknowledgments

This research is supported by the Spanish Ministry of Science and Innovation R&D project ARTEMISA (TIN2009-14378-C02-01).

References

[1] Cook, Diane J., Augusto, Juan C, Jakkula, V. R. *Ambient intelligence : Technologies , applications , and opportunities*. Pervasive and Mobile Computing, 2009, 5(4), 277-??298.

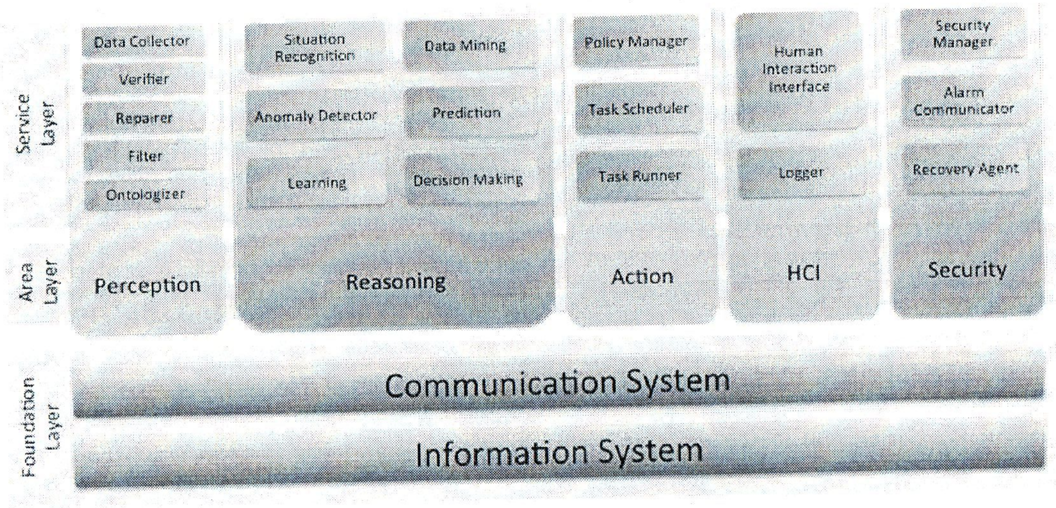

Figure 5: Framework.

[2] Das, S.K., Cook, Diane J. *Designing and Modeling Smart Environments* (Invited Paper). 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks(WoWMoM06), 490-494. Ieee. doi:10.1109/WOWMOM.2006.35

[3] Augusto, J., Nakashima, H. *Ambient intelligence and smart environments: A state of the art.* Ambient Intelligence and Smart Environment, 2007, 1?29.

[4] Zigor Salvador, Mikel Larrea, A. L. *Smart Environment Application Architecture.* Pervasive Computing Technologies for Healthcare. 2008.