Distributed Robotic Architecture based on Smart Services

José Vicente Berná-Martínez, Francisco Maciá-Pérez, Héctor Ramos-Morillo, Virgilio Gilart-Iglesias
Computer Science Department
University of Alicante
Carretera San Vicente s/n, San Vicente, Alicante, 03690
Spain
{jvberna, pmacia, hramos, vgilart}@dtic.ua.es

***Abstract* – This paper presents an approach for designing robots and robotic systems based on the application of models, architectures, techniques and tools that have contributed valid solutions in other areas, such as e-business. Before applying these solutions, the physical elements that make up a robotic system are subjected to a normalization process in order to characterize their functional contributions. In this way, the conceptual model and the technical architecture of the service-oriented architecture robotic system is established. The paper also includes a simple implementation enabling the proposal to be validated, together with the main conclusions reached and future lines of study.**

I. INTRODUCTION

Robotics is an essential component of production automation, affecting labor at all levels from non-specialist workers to professional engineers and production managers. Robotics is a continually growing area, and is subject to great barriers that condition its growth and functionality [1]. Such obstacles include the lack of unicity and standards, even in the most elementary components; the complexity in its design, development and implementation due to the great number of implied disciplines such as informatics, mechanics, electronics, physics and automation; and the great dependency of the underlying technology that makes up and sustains the physical layer.

These types of problems, in which a multitude of actors, disciplines and technologies conjugate themselves, is very common in the development of applications distributed on the Internet and for which information and communication technologies (ICT) have already provided successful solutions. This new scene has forced the business applications to qualify mechanisms that allow their distributed development, reusability, integration of modules and components, and methods of interaction between organizations.

In this way, the ICT, as a whole, have been instituted as the technological bases on which the present industrial and productive weave is sustained. Thus, models, architectures, design patterns and tools that are providing scalable, flexible, integrated and valid solutions, have been developed in the long term.

The technologies which govern the Internet are increasingly permeating the various components of our daily lives. Through concepts such as pervasive computing, miniaturization, intelligent buildings and environments, semantic Web services, intelligent sensor networks and digital skin, ICT are opening up access to new approaches to traditional problems.

Due to the fact that the operative and functional requirements in the area of robotics coincide with those that have been resolved in the area of internet-distributed applications –facing up to the conceptualisation of complex processes which currently involve hundreds of distributed software components, different technologies, interconnection networks, the need for standards, resource integration or collaboration between different systems– we propose applying these same solutions (n-tier architectures, distributed software components, B2B, B2C or M2M models and SOA architectures) to reach a scalable, flexible and realistic model of robots and multi-robot systems that allows us to contemplate each element that makes up a service, independently of its nature, ubication or any other type of physical restriction.

In order to achieve this, we shall put the physical elements which make up a robotic system through a normalization process, enabling them to be characterized by their contribution to the functional model; the technical architecture of the robotic system will be established based on SOA and on open standards; we shall show how traditional robot architecture adapts to the SOA model within the service paradigm; we shall present a simple implementation of a robotic system based on smart services; and finally we shall set out the main conclusions which can be drawn from the study and future lines of research.

II. BACKGROUND

In recent years, robotics has experienced a great advance in the many fields involved [1]. Nevertheless, these advances have not affected all the tasks that a robot may carry out to the same extent. This may be explained by the fact that the models and architectures used were not the most suitable in order to obtain more far-reaching objectives [2]. Aspects such as cognition or commonsense arise from the interaction of individuals, with their context or with other individuals. Therefore, models and distributed architectures that can support these multitudinal ecosystems are required [3]. There are a great number of projects that deal with collaborative systems by which it is possible to conceive the appropriate mechanisms and functionality of independent techniques so that several robots can establish relationships with each other and efficiently obtain the fulfillment of a task [4]. These studies explore the robustness, flexibility and ability to solve complex problems, by using the parallelism and self-organization of robotic communities composed of independent and even heterogenous individuals [5]. Many of these studies propose a technological framework that may serve as a standard platform in the area of robotics [6] and which allows the implementation of the system to be separated from the physical layer [7].

The use of alternative communication mechanisms to replace traditional systems and the study of possible relationships and their formalisation are beginning to be used for this purpose [8]. For many years, fields that are closely linked to robotics, such as sensoring or monitoring, have focused on the use and application of internet-related technologies for the development of intelligent [9] or embedded networks [10], since these technologies are low cost, highly sophisticated, far-reaching and socially acceptable. Furthermore, they are now the main tools for enabling distributed infrastructures to be established and to overcome barriers related to physical technologies [9].

On the other hand, the use of ontologies arises as a mechanism to standardize and formalize the interoperability of current systems, presenting itself as a means of representing our shared knowledge in a specific domain, and commonly associated with the notion of semantic containers [11].

The miniaturization capacity allows us to incorporate computation in practically any component [12]. The concept of pervasive computing along with communications technologies can be applied to all areas, for example, using encrusted devices to provide internet-based interfaces as a device management mechanism [13]. Some of the latest studies propose intelligent environments combining the use of embedded sensors and ontology-based contexts [14]. The success of embedded computation is evident due to the transparent incorporation of ITC into daily life [15].

III. NORMALIZATION OF COMPONENTES

One of the main problems with robotics is having to approach the heterogeneity existing in the different physical devices on all levels. It would be desirable for these physical devices to be as homogenous as possible, at least conceptually. This would enable the border between all the disciplines involved to be clearly defined and would provide a unified vision of the elements, thus simplifying the model of complex robotic systems.

The aim of the normalization process is to characterize the elementary components of a robotic system (including sensors, actuators and computational elements) from the point of view of its contribution to the robot's functional and conceptual model. In this way, a vision is generated that firstly allows physical elements and, later, robotic processes to evolve towards ITC services. The process involves equipping each of the robot's elementary components with the required hardware infrastructure and software so that they can be displayed as software services. The new resulting components of this process are named Smart Services and allow us to raise the abstraction level of the lowest layers, until they can be compared with the rest of the robotic system's software services.

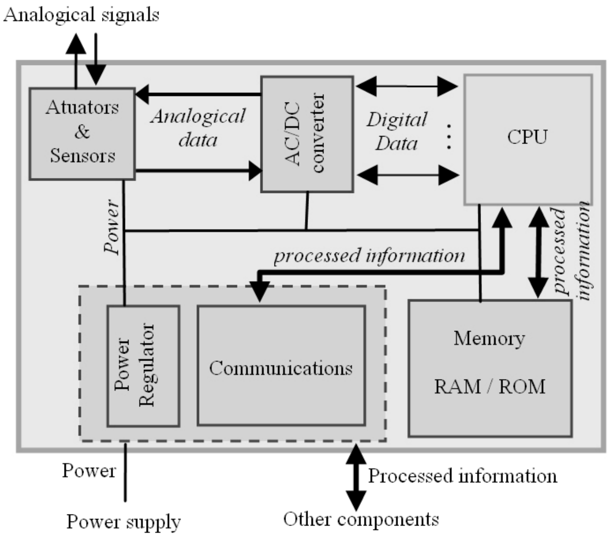

Figure 1: Physical scheme for a smart component hardware

More specifically, we will firstly act on the sensors and actuators. In general, these elements do not have the capacity to process and store the information with which to operate. The first stage involves equipping these elements with the necessary hardware so that they can do so. Figure 1 shows a block diagram with the main components.

This hardware consists of: an analogical-digital converter that allows the analogical signals of the actuators or sensors to be adapted to those of the digital processor; a processor, equipped with computation capacity; memory to store or handle the data obtained or processed; a communications system that adapts the information to the communication channel; and an energy regulation module that adapts input at the correct levels. The latter two modules could be integrated since existing communication protocols can unify these functions (PoE or PLC).

With the incorporation of computational capacity and storage into the device, the elements are transformed into cognitive components (i.e. they send and receive messages which they process and interpret), with a standard communication capacity. We thus have a computational platform which enables these devices to be processed from another perspective.

Although we now have a much more powerful base, we have to bring it within the standards which interest us, for which we turn to techniques which enable us to build in a platform which integrates the protocols and appropriate services (TCP/IP, SOAP, XML, etc.).

This platform consists of: a communications layer with the standard network protocols (TCP/IP, HTTP, SMT, FTP...); a second layer where SOAP and its extensions are located; a third layer with the service description languages, WSDL; and a fourth layer where the discovery services and publication services (UDDI) will be located. The three upper layers of the platform use technology bases such as XML, DTD and schemas. The whole platform is covered by the standard network security

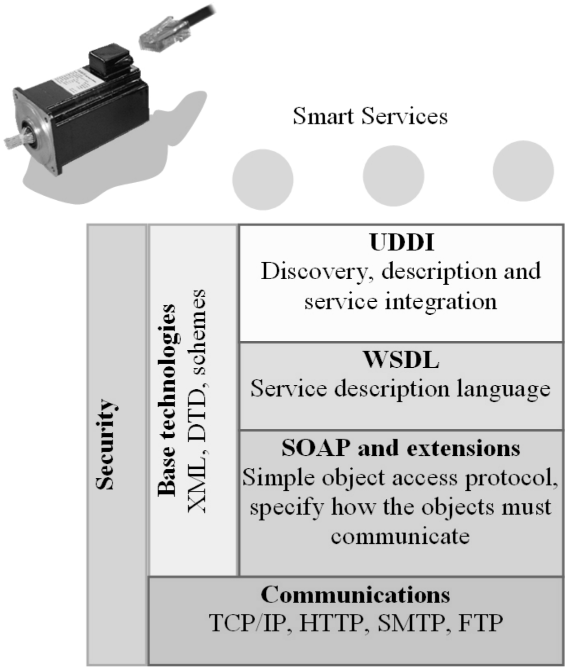
Smart Services
UDDI
Discovery, description and service integration
WSDL
Service description language
SOAP and extensions
Simple object access protocol, specify how the objects must communicate
Communications
TCP/IP, HTTP, SMTP, FTP
Security
Base technologies
XML, DTD, schemes

Figure 2: Smart Service logical scheme

protocols. In this way, the services offered by each component will be found in the upper levels.

These new devices, with their capacities, are what we known as Smart Sensors or Smart Actuators.

In addition to sensors and actuators, we require the rest of the computational components that make up the robotic system to be aligned technologically and functionally with these new intelligent devices. Computational elements are those responsible for carrying out calculations, taking decisions and implementing the policies and strategies necessary to achieve the objective. In this case, the required computational hardware platform is already available. For this reason, it will be sufficient to add the necessary architectonic software layers so that planners, gateways, trajectory calculation processes, controllers and other functionalities of the robot can also be offered as services. This type of services that originate from software processes are known as Smart Computations.

Finally, we have been able to encapsulate and hide the different physical elements involved in the system and, since they are all now shown as services, we have grouped them under the common name of Smart Services, independently of their physical nature.

Given that now there is nothing to prevent a Smart Service from being made up of other Smart Services, we can distinguish between Basic Smart Services, such as those which cannot be divided into other Smart Services, and the Compound Smart Services, which use at least one other Smart Service, which in turn may be either basic or compound.

IV. CONCEPTUAL TECHNICAL ARCHITECTURE

Once all the elements have been standardized and reduced to distributed software components (Smart Services), we can apply the solutions extracted from the e-business models or, more generically, from the distributed software component-based systems.

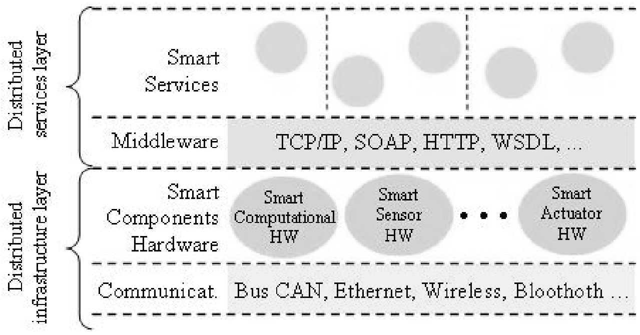
Distributed services layer
Smart Services
Middleware
TCP/IP, SOAP, HTTP, WSDL, ...
Distributed infrastructure layer
Smart Components Hardware
Smart Computational HW
Smart Sensor HW
• • •
Smart Actuator HW
Communicat.
Bus CAN, Ethernet, Wireless, Bloothoth ...

Figure 3: Conceptual technical architecture for a distributed robotic system based on Smart Services

In the case of the technical architecture of the robot, an n-tier architecture can be applied where a homogenous and structured panorama is formed in layers. A communications level is determined to support the Smart Service hardware, which is formed by the whole physical layer which makes up each of the components. The Middleware layer is comprised of the software that allows us to manage the described communications and processing mechanisms. It will be composed of the transmission protocols, messages language, component access protocols, discovery protocols, etc. This level is responsible for abstracting the traditional components towards the world of the services. As regards the technical physical architecture, each intelligent hardware component, together with its service layer, constitutes a container upon which are executed the distributed software components that implement their business logic or, in terms of the robotic system, the functionality of their components. The service layer is the platform for developing the software components that provide the functionality and interface that the Smart Service is able to offer to other components, whether they belong to the same robot or to any other client with the sufficient capacity and permission to call it

The resulting architecture is known as Service-Oriented Robotic Architecture and incorporates characteristics such as the organization of the elements involved into perfectly defined compartments and interfaces.

The elements which comprise the traditional architecture of the robot (fig. 4a), once transformed into Smart Services, will operate as service providers, service consumers or even as both, reflected perfectly in the new architecture.

This produces a notable increase in the necessary technology (fig. 4b) in service-oriented architecture, since the services have to be supported by a series of technologies and protocols. This means that the model is more complex than that of traditional robotics, although with the advantage that standard and open platforms already exist for its implementation, with all the characteristics of the SOA paradigm transferred to the robotics.

Each component has been transformed into a method or high level object, with a stable, public interface, which can

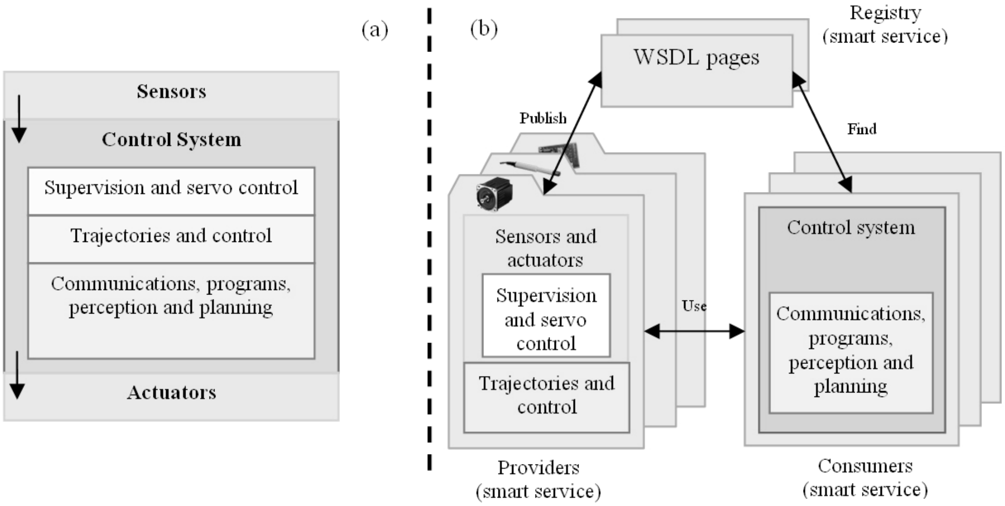
(a)
Sensors
Control System
Supervision and servo control
Trajectories and control
Communications, programs, perception and planning
Actuators
(b)
WSDL pages
Registry (smart service)
Publish
Find
Sensors and actuators
Supervision and servo control
Trajectories and control
Use
Control system
Communications, programs, perception and planning
Providers (smart service)
Consumers (smart service)

Figure 4: (a) Traditional robotic architecture. (b) Service oriented robotic architecture

be called upon by any client and developed independently of the system in which is it to be implanted. It enables the homogenization of the actors involved and a standardization of the interfaces, methods and technologies involved in both development and use, thus facilitating portability and keeping costs down.

Increased computational capacity and layering of protocols and business logic over Smart Services deliver a distributed middleware, based on standard technologies and independent of manufacturer.

Each component possesses intelligence, can be consulted or involved in an action and brings to the system its description, its availability and interfaces, and the organization of its output data or input parameters, thus enabling the establishment of relations within a Smart Services community, instead of being isolated elements.

The range of communication platforms is growing, since we can use whichever SOA model support is best adapted to the system requirements; this also enables the incorporation of their respective security mechanisms.

The knowledge base and tools of the ICT community are freely accessible, so that the benefit can be gained of all the work carried out to date in the development of Web Services as mechanisms for reusing designs and components, service modeling, component integration etc..

By elevating the technological level of each component involved in a robotic system, we are able to ensure that the levels of functionality which have to be reached for the development of a project are also raised. When a project is unfolding, those responsible for it are thus able to focus their attention on the relationships which have to be established between components and not on how communications between them should be achieved, what type of access is required, response times, physical aspects of the devices, signal processing or other such basic details. This leads to less overlap between the disciplines involved and efforts can therefore be concentrated on the development and modeling, rather than on the physical architecture and technology involved, thus facilitating the specialization of each person involved in a project and the creation of multidisciplinary teams.

The higher technological level enables the stages of design and of implementation to be closer, since for a particularly abstract design we can use components which are also high level objects, based on pre-existing, widely accepted frameworks.

The system will consist of independent models which will gradually be implemented into the system, since the modularity, scalability and integration capacity are implicit in the service architecture, enabling the reuse of designs and components in the construction of diverse robotic systems.

V. USE CASE

The scenario which is considered as a framework for the study comprises an Ethernet network in which a PC will be housed, containing the heaviest software components, responsible for planning and control, and a server for the UDDI services register, interconnection with other networks and a database. Our robotic system will consist of various basic services communicating between themselves via an Ethernet network and the group can be linked up to the system network through a WIFI access point so that our system would have complete mobility without losing the communication channel.

Since we are in the initial stages of the study, we shall implement a basic smart service which offers us flexibility for testing. For its construction will shall use a programmable RCX brick and an embedded XPort device.

The programmable Lego RCX brick has a Hitachi H8/3292 microcontroller embedded, with an H8/300 CPU Core which runs the sensor control and actuator programs, and an infrared port which we shall connect to the serial port of the XPort device, thus converting it into a device accessible via conventional Ethernet. XPort is a compact solution which enables a component to be web-enabled and incorporates memory and its own processor, capable of executing the necessary protocols and software. A tension regulator will enable both devices to be powered simultaneously (fig. 5).

XPort will be responsible for receiving SOAP messages,

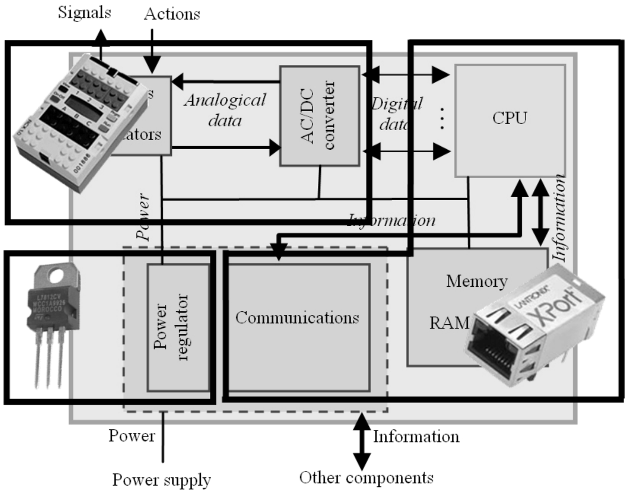

Figure 5: Hardware implementation using RCX y XPort

interpreting them and sending the orders to the RCX to interpret and to take the necessary actions. The real computational nucleus thus lies in the XPort, which will house the logical level, while the RCX brick will simply act as an analogue/digital converter to interact with the physical devices.

The Smart Service obtained, inheriting the characteristics of the RCX brick, will be able to interact with three inputs and three outputs simultaneously. As set out in the corresponding WSDL sheet, we shall only house a service in this device using two methods, one which will serve to order an action of an actuator and the other which will check the state of a sensor. This service will be registered in the UDDI server which we have put on the network so that any user may access and utilize it. In this case we shall create a simple client which just uses these services for small movements, so that we are able to confirm the functioning of the Smart Services based on SOAP. Using the flexibility of Lego, we shall create a small system comprising a mobile platform and a contact sensor, each controlled by its own Smart Service.

VI. CONCLUSIONS

In this paper we propose a conceptualization that breaks down the architecture of the underlying technology and frees the traditional organisational schemas of their limitations; it allows us to disconnect the disciplines involved in the development of projects by elevating the functionality of the minimum components, thanks to the appearance of middleware; and it homogenizes the technologies that will be used when separating and organizing the different functional aspects in layers and levels with well-defined interfaces.

The benefits not only affect the architectonic aspects, but they also allow us to take advantage of the conceptual and organisational characteristics of the service paradigm, ensuring a simpler integration based on standards, scalability, regardless of the platform and manufacturer, and using realistic development tools. The technological gap that separates implementation models is reduced as since the minimum elements of the robotic systems are technologically more advanced..

We have presented a simple implementation of Smart Services, demonstrating that the use of current tools is feasible and that it is possible achieve the move, in simple form, from the design of a system to its realization thanks to the high technological content of its basic components and the use of high level tools.

The proposed approach opens the door to acquiring new capabilities such as self-repair, self-assembly or service replication, since the technologies that sustain the service paradigm can support or implement mechanisms related to these proposals.

Finally, we are currently working on a formal framework for the modelling of robotic and multirobot systems, based on the formal characterization of the elements involved and the study of the relationships between these through the use of ontologies.

VII. REFERENCES

[1] Torres, F., Pomares, J., Gil, S., Aracil, R. "Robots y Sistemas Sensoriales". Prentice-Hall. 2000.

[2] Minsky, M. L.. "DRAFT. Future Models for Mind-Machines". Proceedings of the AISB'00 Symposium on Creative & Cultural Aspects and Applications of AI & Cognitive Science. Birmingham. pp.41-53. 2000.

[3] Oatley, K. "Shakespeare's invention of theatre as simulation that runs on minds". Proceedings of the AISB'00 Symposium on Creative & Cultural Aspects and Applications of AI & Cognitive Science. pp.41-53. Birmingham, UK, 2000. ISBN 1 902956

[4] Grob, R., M. Bonani, M., Mondada, F., Dorigo, M. "Autonomous Self-assembly in a Swarm-Bot". Proceedings of the Third International Symposium on Autonomous Minirobots for Research and Edutainment. Springer. Verlag. 2006.

[5] Mondada, F., Pettinaro, G., Guignard, A., Kwee, I., Floreano, D., Deneubourg, J., Nolfi, S., Gambardella, L., Dorigo, M. "Swarm-Bot: a New Distributed Robotic Concept". Autonomous Robots. Vol 17, 2004. ISSN: 0929-5593. pp. 193-221

[6] Spears, W., Gordon-Spears, D., Heil, R."Distributed, Physics-Based Control of Swarm of Vehicles". Autonomous Robots. Vol. 17, 2004. ISSN: 0929-5593. pp. 137-162.

[7] Pérez, A. "Of implementing neural epigenesist, reinforcement learning, and mental rehearsal in a mobile autonomous robot". Proceedings of the AISB'00 Symposium on Creative & Cultural Aspects. 2000.

[8] Sekmen, A. "Multi-Robot Communication via Grid Services". IASTED International Conference on Robotics and Applications. Honolulu. 2004.

[9] Delin, K., Jackson, S., Johnson, D., Burleigh, S., Woodrow, R., McAuley, M., Dohm, J., Ip, F., Ferré, T., Rucker, D., Baker, V. "Environmental Studies

with the Sensor Web: Principles and Practice". Sensors. Vol. 5, 2005. pp 103-117.

[10] Tao, V., Liang, S., Croitoru, A. "GeoSWIFT: Open Geospatial Sensing Services for Sensor Web". en A. Stefanidis and S. Nittel (ed.), GeoSensor Networks. CRC Press. 2004. ISBN: 0415324041, pp. 80-86.

[11] Sánchez, D. M., Cavero, J., Marcos, E. "Ontologías y MDA: una revisión de la literatura", en Estévez, A., Pelechado, V. y Vallecillo, A. (ed.) Actas del DSDM05. 2005.

[12] Gilart, V. , Ramos, H., Maciá, P. "Del eBusiness al Smartscrew" en Maciá, F. y García, J. M. (ed.), Servicios Electrónicos para la Sociedad de la Información. San Vicente del Raspeig. Publicaciones de la Universidad de Alicante. 2006.

[13] Ju, H. T, Choi, M. J., Hong, J. "An efficient and lightweight embedded Web server for Web-based network element management". International Journal of Network Management. Vol. 10, num. 5, 2000.. ISSN:1099-1190. pp. 261-275

[14] Tan, J. G., Zhang, D., Wang, X., Cheng, H. "Enhancing semantic spaces with event-driven context interpretation". En Gellersen, H. W., Want, R. y Schmidt, A., Pervasive Computing. 2005. LNCS 3468. ISSN: 0302-9743, pp. 80-97

[15] Hansmann, U. , Merk, L., Nicklous, M., Stober, T. "Pervasive Computing". Böblingen, Germany. Springer. 2003.