IntechOpen

Ambient Intelligence

Edited by Felix Jesus Villanueva Molina

Ambient Intelligence

Edited by

Félix Jesús Villanueva Molina

Ambient Intelligence
http://dx.doi.org/10.5772/194
Edited by Felix Jesus Villanueva Molina

First published in Croatia, 2010 by INTECH d.o.o.
eBook (PDF) Published by IN TECH d.o.o.
Place and year of publication of eBook (PDF): Rijeka, 2019.
IntechOpen is the global imprint of IN TECH d.o.o.
Printed in Croatia

Legal deposit, Croatia: National and University Library in Zagreb

Additional hard and PDF copies can be obtained from orders@intechopen.com

Ambient Intelligence
Edited by Felix Jesus Villanueva Molina
p. cm.
ISBN 978-953-307-078-0
eBook (PDF) ISBN 978-953-51-5883-7

Preface

In 2001 ISTAG published “Scenarios for Ambient Intelligence in 2010” which describes a set of scenarios showing the interaction between the users and the information and communication technologies. We can affirm now and with our actual perspective that maybe they were too optimistic but the Ambient Intelligence vision described in this document is still valid and any researcher can take these scenarios as a guideline in his research activity.

We cannot deny that Ambient Intelligence concepts are moving from research labs demonstrators to our daily lives in a slowly but continuous way. However, we are far from concluding that our living spaces are intelligent and are enhancing our living style. Ambient intelligence has attracted much attention from multidisciplinary research areas and there are still open issues in most of them. In this book we analyze and study a selection of open problems which we consider the key ones in order to let ambient intelligence become a reality. We expect that this book provides the reader with a good idea about the current research lines in ambient intelligence, a good overview of existent works and candidate solutions for each one of these problems.

Specifically, chapter 1 and 2 are devoted to a recurrent problem in ambient intelligence, the integration of heterogeneous devices, technologies and services deployed in any scenario following the ambient intelligence vision. Next, in chapter 3, we focus on indoor localization technologies in real environments since we have the conviction that more advanced services can be provided with an exact information about user positions. In the same line, in chapter 4 we present a visual analysis framework of user behavior in ambient intelligence environments. This type of applications are the seed forenabling service adaptation to the user activity on real time. In chapter 5 the interactions between the user and the ambient intelligence environment by means of dialogue systems are analyzed. This type of systems represent the most natural way of interaction for the users. Finally, ambient intelligence environments will only be successful as far as they can be trusted by both users and services. Chapter 6 tackle this problem by modeling properties of trust concept in general computing systems.

Each one of these chapters have been written and revised by experts in each topic which is a guarantee of quality in such way that the potential audience of this book, students and researches who want to specialize in ambient intelligence field, can find an excellent analysis of each of the topics tackled in each chapter and a description of author’s proposed solutions.

Finally we would like to express our gratitude to all authors for their contributions and also to the reviewers for their time and corrections.

Félix Jesús Villanueva Molina

Contents

1

Integrating Ambient Intelligence Technologies Using an Architectural Approach

A. Paz-Lopez, G. Varela, S. Vazquez-Rodriguez, J. A. Becerra and R. J. Duro
Grupo Integrado de Ingeniería, Universidad de A Coruña
Spain

1. Introduction

It seems difficult to agree on a concise and accurate definition of a concept as broad as Ambient Intelligence. In short, it could be said that in such a context people interact with each other and with the environment and are assisted in their tasks in a certain smart way. The term Intelligence concerns the response that users expect from the system in terms of what Artificial Intelligence stands for, that is, proactivity, predictability and adaptability in its behaviours. This behaviour collection constitutes the functionalities that characterize an AmI system. On the other hand, the term Ambient is related to human factors and conducts and the ubiquitousness of a non invasive system. Indeed, the main goal should be user expectation fulfilment and the dynamic adaptation of the system to his/her needs, preferences and habits. In this context, the environment must be understood as an extended environment where a person works, has its family life, its recreational time and its social life. Therefore this environment involves the workplace, the home, the car or public places such as malls or sports centers, among others.

At this point, some questions arise: How should so vast and dynamic environment be managed? Where should the different software/hardware components be located? How do these components interact each other? How are new hardware/software components incorporated to an AmI system and how is it rearranged in order to integrate those elements? What are the channels through which information flows and who should supervise them? How could we take advantage of the numerous information sources/targets, like sensors, actuators or microprocessor appliances, and bus technologies available in a certain context? The idea that underlies all of these questions is the concept of an Architecture that should provide support and consistency to hardware/software AmI components and should allow AmI functionalities to be developed and integrated. This chapter deals with what an architecture conceptually involves and the description of possible approaches to the different issues related to it.

An architecture implies aspects concerning low-level hardware access, middleware and architectural software considerations. Therefore, it doesn't provide any AmI functionalities by itself but, on one hand, provides tools to system designers for the rapid development, deployment, reusability and interoperability of AmI solutions; On the other hand, it serves

as a technological wrapping that grants coherence and compatibility between AmI components contained in it by defining a specific model for each element.
At a first glance the approach of using the general architectural model would seem to be a good beginning in order to try to solve the problems indicated before. There is, however, no real consensus on this point in the literature. Many researchers just emphasize the development of AmI functionalities while hardware/software structural issues are pushed into the background. Hardware access matters and the complexity of physical network structures are often underestimated. Also information concerning AmI services cannot usually be shared with others and, currently, most AmI components are not aware of the existence of other components within the same system.
Despite this conundrum of approaches and solutions, it seems clear that the design of a global AmI architecture is necessary if good software/hardware engineering practices are to be achieved and the systems show some of the properties desired such as scalability, fault tolerance or ease of deployment and integration. It is in this line that some researchers have opened up the field to AMI architecture design. It is the case of the SOPRANO (Wolf et al., 2008; SOPRANO, 2007) and PERSONA (PERSONA, 2008) projects where platforms are based on OSGi and SOA (Service Oriented Architectures) were employed. From another point of view, a multi-agent based platform is proposed by AIT (Athens Information Technology) through the CHIL (Soldatos et al., 2007) project. The AMIGO project proposes a combination of both technologies, that is, SOA and agent based in (Vallee et al., 2005).
We will start this chapter by identifying the multiple topics that an architecture involves and the needs that come up when developing real solutions for them. As a result of this first analysis, we will define the conceptual model of an AmI architecture that is dealt with in next section. It is structured as a collection of subsections concerning the different model challenges. For each one of them an analysis of interest and needs, a review of the state of the art and a description of our approach as compared to others are presented. In short, structural software matters concerning the software platform, resource cooperation, knowledge representation and ubiquity are dealt with as well as user-system interaction, hardware virtualization and artificial intelligence functionalities. All the proposals that we describe here were developed in the scope of a global architectural project that we know as HI3 Architecture. After this, we will consider an example of using an HI3 system. The usefulness and the interoperability of the modules described will be shown with the aid of simple cases. Some conclusions and further avenues of research are outlined at the end of the chapter.

2. Needs and challenges

There are many requirements in the design and development of an AmI architecture. Some of them are common in distributed systems (with or without Artificial Intelligence), whereas others are very specific of AmI environments.
First of all, it is obvious that the whole system will be made up of many different elements: sensors, actuators, applications providing functionalities to users, intermediate services that provide common functionalities to applications, system level utilities for system control and logging, etc. All of these elements need to communicate to each other, consequently, communications are our first clear and important need. Due to the fact that some components, mainly sensors and actuators, will be physically distributed, special attention

will have to be paid to communications in order to guarantee reliability. Communications should also be optimized in two aspects; minimize resource consumption, as some components will have to run in devices with much reduced computational power, and minimize processing time, because some applications will need real time, or near real time, capabilities. Some sensors are susceptible of requiring high throughput, video cameras for example, so the system should also support such high level throughput. In addition, every AmI system should have the capability of growing, it is quite frustrating to try to expand an existing installation and find a limit in the number of elements or throughput (as is usually the case in current commercial systems). Thus, scalability is almost a must, and this brings together the ability to process a large volume of data in communications. The final characteristic related with communications is security. Information managed in an AmI system is usually sensitive, as it could be used to describe private aspects of its users. Therefore, communications should be secured through encryption and authentication. Security should also be included at a component level, in order to protect malicious replacement / injection of elements in the system, and at the user level, for obvious reasons.

Reliability is not only applicable to communications, it is a general requirement of any AmI system, due to their nature: AmI systems have to interact with human beings in a non-intrusive way, and it would be unacceptable for the user to have to restart an AmI system when something fails. That is, reliability is necessary to achieve high availability. In this context, redundancy is fundamental, although not sufficient, to obtain reliability. It means that if any software or hardware component fails, its functionality is immediately replaced by other components. Another characteristic that is necessary for reliability is fault tolerance. Eventually, everything can (and will) fail, consequently hardware and software must be designed to be able to recover when that happens. Related with this, some kind of hardware hot-swap is, obviously, very interesting.

From a software engineering point of view, all the elements should be designed bearing reusability in mind. The main reason is that an AmI system can, potentially, grow throughout its operational life, and new needs should be solved as soon as possible reusing components that are already available to the largest extent possible. In order to maximize the possibility of reusing components, system elements should be as modular as possible. Related to this and to high availability, it would be interesting to have also software hot-swap capabilities, so that a given element in the system can be replaced without interfering with the other elements. So as not to be tied to a specific hardware / O.S., something that is not desirable as it would reduce application scope, the software employed and programmed should be multi-platform, making hardware / software dependencies as low as possible in terms of software layers. Other related thing to take into account is that, if we really want hardware independence at sensor level, the AmI architecture should support different protocols of different hardware makers, trying to abstract these different protocols as soon as possible in the layer hierarchy. Finally, it is necessary to have some tools to control, manage and inspect the behaviour of an AmI system.

Human activity is complex, highly unpredictable and situation dependent. Unobtrusive applications within these intelligent spaces need information about the current state of the environment to dynamically adapt to different situations. Adaptation and unobtrusiveness are essential to social acceptability of AmI systems. These characteristics of human environments make it impossible to pre-program the appropriate behaviours for a context-aware service or application. Hence, an AmI architecture should include an extensible,

adaptable and efficient model for handling and sharing context information, providing a structured and integrated view of the environment in which the system runs.
These requirements and needs really condition the way an AmI architecture should be designed. They are obviously general principles that have to be taken into account from the beginning when considering the design of the main components and structure of the architecture. Many more needs and requirements will arise as we go deeper into de details of each component, and we will comment on them in the corresponding sections.

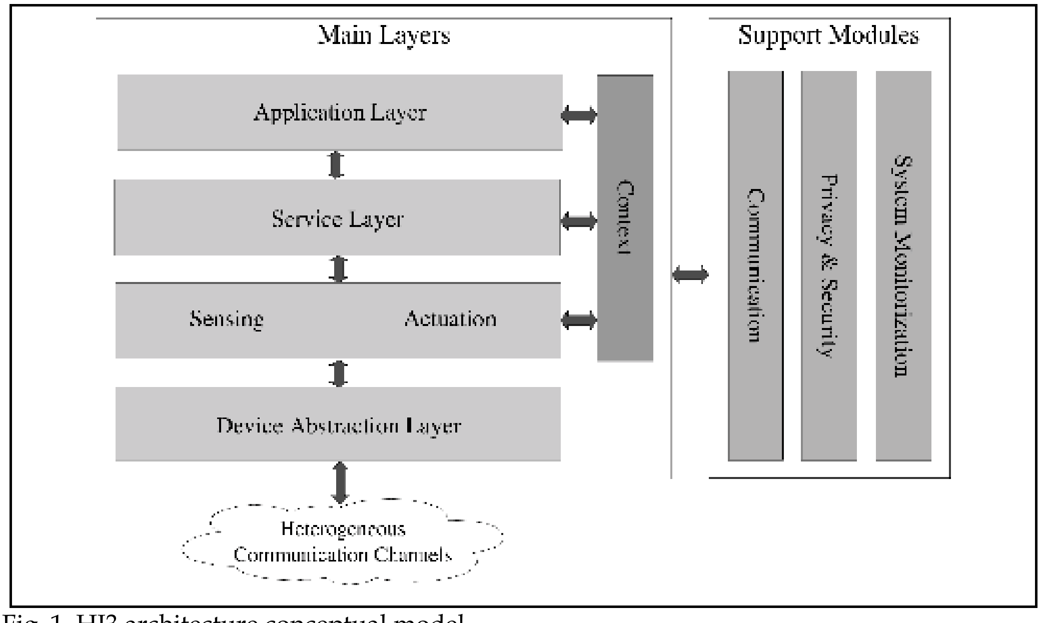

Fig. 1. HI3 architecture conceptual model

3. Architecting ambient intelligence

In the previous section we have identified a set of properties or key areas that we think an integral Ambient Intelligence software development platform should tackle in order to alleviate the problems associated to AmI software development. In this section we are going to address thewse diverse topics, providing a brief review of the most interesting approaches found in the literature to provide solutions to each of them.
Along with the detailed description of each key area and its state of the art review, we will present our version of an integral general purpose AmI system development architecture,. This allows us to provide a complete vision of the AmI software development field, going from the problems and requirements of the developers to the construction of a complete solution for them.
Before starting the discussion about the different key topics, and in order to facilitate the description of the different areas of the proposed solution, we will give here a brief overview of our approach. In the last few years we have been working on the development and application to different environments of a multi-layer intelligent agent based software architecture called HI3 (Varela et al., 2007; Paz et al., 2008, a; Paz et al., 2008, b). HI3 stands

for Humanized, Intelligent, Interactive and Integrated environments. The objective of this architecture is to provide the methodology, structure and base services to easily produce modular scalable systems in ambient intelligence settings that can operate without downgrading their performance as the system grows. The main idea behind our proposal is a layer based design that enables the division of system elements into levels, reducing the coupling between modules, facilitating abstraction as well as the distribution of responsibilities. A high level view of the design proposal is shown in Figure 1. A further description of this design will be presented in following sections and more information can be found in (Varela et al., 2007; Paz et al. 2008, b).

In order to implement this conceptual model, we have chosen a multiagent technology based approach supported by the JADE agent framework. Many of the characteristics of this paradigm fit perfectly with the main objectives for our architecture, such as autonomous operation, distribution and collaboration on task resolution, proactivity, intrinsic communications capabilities, mobility and adaptability.

3.1 Conceptual modeling

AmI systems are much more complicated than traditional computing systems. Hence, characteristics such as adaptability, flexibility, interoperability and modularity are more important. Furthermore, these systems must provide common improvements such as service discovery, self-organization, rich knowledge representations and context-awareness. To address these challenges, especially when seeking real AmI system interoperability, a general reference conceptual model and architecture is necessary. In recent years, a lot of research institutions and universities have been working on the definition and development of architectures and middleware that deal with the complex characteristics of AmI environments. However, only a small part of these projects try to formally define conceptual models that are not restricted to particular AmI domains or environments.

Some recent research efforts in the direction described above can be found, for instance, in (Rui et al., 2009) where the authors defend the need of general reference models in AmI and present a conceptual hierarchical model to describe a typical AmI application system. They propose a theory to describe AmI systems using a five-layer model: sensors and actuators layer, AmI network and middleware layer, devices layer, service layer and AmI applications layer. They do not include, as part of this work, high-level development tools that could closely guide developers in the process of building systems that fulfill the proposed conceptual model. Another example is the PERSONA project (PERSONA, 2009) which focuses on the specification of a reference architecture for ambient assisted living (AAL) spaces. The authors formally define a logical architecture that abstracts all functionality not leading to context, input, or output events as services.

Inspired by some of the ideas underlying these projects we developed the HI3 architecture, focusing on interoperability, scalability and ease of deployment.

HI3 Architecture conceptual model

As previously described, the HI3 architecture proposed here has a conceptual structure based on a multi-layer design schematized in Figure 1. This conceptual model establishes criteria and defines recommendations that facilitate developers the division of AmI systems into decoupled modules as well as the distribution responsibilities using homogeneous design principles and patterns. Communications between layers is vertical. Within layer, horizontal communications are allowed thus permitting the cooperation between elements. The main layers defined in this model are:

- Device Access Layer: It includes elements that encapsulate concrete logic to access physical devices. It is divided into two sub-layers: device drivers and device access API.
- Sensing and Actuation layer: There are two types of elements in this layer: sensing elements and actuation elements. Sensing elements provide access to sensor type devices (using the device access API). Similarly, actuation elements must provide the required functionality to command actuator type devices. The elements of this layer are proxies of the physical devices.
- Service layer: A service is an element designed to solve a high level task that does not provide a complete solution for a system use case. Services are managed through a repository that enables the discovery of services required by others to perform their task.
- Application layer: This layer hosts the elements representing and implementing particular functionalities that a user expects from the system. These components make use of services registered in the system to carry out their tasks.

Furthermore, there is a component that is shared by the three higher level layers, the context. Its main goal is to define and manage the elements that must be observed in order to model the current state of the environment.

The previous high-level conceptual model provides a common vocabulary and guidelines to make the construction of AmI systems easier through technology abstraction and component composition and reusability. However, we believe that this model must be supported by middleware that abstracts developers from repetitive tasks and complex interactions with particular technologies. Specifically, we chose a multi-agent based approach to develop a middleware solution (fulfilling the hi3 architecture conceptual model requirements) that builds up a container for AmI applications providing advanced features such as high level inter-agent communications facilities, multi-agent models with support for the declarative definition of components, distributed management tools or development utilities (see section 3.2 for technical details).

This multi-agent based middleware promotes the division of large and complex AmI systems into highly decoupled components and, at the same time, provides what we call a multi-agent declarative model. The multi-agent declarative model is, on one hand, a model to describe a multi-agent system and its components and, on the other, it is a set of tools to declaratively define these components. As a model it brings together all the different elements that our platform leaves in the hands of the developers to build new applications. The top level concept of our platform is the multi-agent system (MAS). The platform is prepared to support several types of MAS with different high level characteristics but, in its basic form, they are a collection of instances of different types of agents that work together to perform a task. Directly below the MAS concept is the agent-type concept. An agent-type specifies the components that will define an agent in our platform, providing a way to declaratively build new agent-types by reusing pre-existing components and mixing them with new ones. Agent-types are defined according to a set of concepts that also have independent and declarative definitions: behaviours, message processors, publications, arguments for agents parameterization and descriptions of public functionalities.

Together, the high-level conceptual model and the multi-agent declarative model provide common guidelines and tools that control the entire process of developing interoperable AmI systems.

3.2 Middleware fundamentals

As previously stated, sharing a common vocabulary, concepts and abstractions among AmI developers is the first step towards more interoperable, reusable and easier to design systems. In the end it brings to the AmI field what UML or the object oriented programming paradigms brought to the programming field. Like paradigms such as OO, a conceptual model for AmI systems by itself does not produce a huge improvement on AmI development efforts. Developers still need to implement those designs that will be heavily based on complex distributed operations and on the integration of a huge number of techniques and technologies. In order to speed up the development of AmI systems, developers will need software, or more precisely, middleware, that supports the previously described models and provides them with tools and utilities that ease the development by allowing them to focus on the logic of the desired functionality, rather than on the interaction and integration of the different underlying technologies and devices.

As AmI is a relatively new field (it started to capture attention on the last ten years) until recently, the majority of projects had their focus set on achieving functionalities for very concrete niches (some examples may be found in (Tapia et al., 2004; Brdiczka et al., 2009; Krumm et al., 2000; Bernardin et al. 2007), taking ad-hoc design and implementation decisions focused on the particular environment dramatically reducing the possibility of adapting those projects to other environments, objectives or technologies. In order to operate in a real life scenario, every one of those projects needed solutions for very similar problems, like distributed component deployment and communications, sensor/actuator hardware interaction or fault tolerance. To put an end to that process of continuously reinventing the wheel, much in the same way as had previously happened before in other areas like multi-agent systems, numerous projects have been started with the aim of producing general purpose middleware that make the development of AmI systems easier.

From a very superficial point of view, it is possible to classify them according to their objectives. On one hand there are projects that are focused on delivering solutions to specific environments, they make decisions and adopt technologies that may make them less general purpose but easier to use in those environments. Prominent examples are projects like SOPRANO (Wolf et al., 2008), focused on elderly care scenarios, (and which relies on abstractions from that environment for component definition and on an ontology based semantic system for component interaction). The already mentioned PERSONA project devoted to increasing the independence of old people or the iRoom (MIT Oxygen, 2009), centred on the work environment. On the other hand we may find projects with a general purpose in mind. These projects usually devote great efforts to technology interoperability aspects and to providing high level abstractions and communications capabilities suitable for the typical requirements of AmI systems. Examples are the AMIGO project (AMIGO, b, 2009), although mostly focused on the home entertainment environment it provides a very versatile architecture, the EMBASSI project (EMBASSI, 2009), that tries to provide the needed technological infrastructure for the development of AmI systems in environments like the home, automotive vehicles and kiosks. The works carried out by Chen Rui et. al (Rui

et al., 2009), towards the definition and implementation of a general purpose AmI framework and our own approach, the HI3 project (Varela et. Al, 2007; Paz et al., 2008, b).
Before starting the discussion on the different technical approaches to the development of ambient intelligence middleware, it is important to take a look at the objectives of AmI in order to establish a series of prerequisites or key areas that a middleware for AmI development should deal with. One key differentiation factor of AmI with respect to general software development is that AmI systems are intended for operation in the real world, interacting with the environment and its inhabitants, this fact alone brings in some strong requirements such as:

- Multi-platform compatibility and interoperability.
- Modularity.
- Distributed operation.
- Dynamic management of components.
- System management.

Multi-platform compatibility and interoperability
AmI software needs to support its deployment on very different software and hardware platforms, from PCs to mobile phones, going through home appliances, automotive vehicles or consumer electronic devices. Because of this almost all the projects having to do with the development of AmI middleware are implementing them with technologies that inherently support multiple platforms, like interpreted programming languages such as Microsoft .Net or SUN Java and XML based mechanisms for communications.

Modularity
Most authors agree that AmI systems need to be highly modular in order to model and interact with a highly dynamic and complex environment like the real world. They need to contemplate and interoperate with a large number of devices and technologies to sense, reason and act on the environment. A design based on loosely coupled components is essential to accommodate that disparity in a sustainable and scalable system. It also forms the basis for component reusability, a key capacity to make more extensible and adaptable systems.
Modularity is a key aspect of AmI middleware, this can be seen in the fact that the projects we cited before can be divided by the techniques they apply to define components. Two main approaches can be distinguished. Projects that rely on Service Oriented Architecture models (SOA), and projects that use a multi-agent approach.
In the last decade, SOA has been widely adopted in the enterprise development field as a good approach for obtaining more sustainable and scalable systems. The SOA approach promotes a system design based on services (components) that define an interface that other components can use to access the functionality of the services without knowing anything about the service´s internal workings. The key design principle of SOA is that the services must be highly decoupled and reusable so that systems can be built by the aggregation or orchestration of multiple services. Numerous AmI projects have adopted SOA as their basis for system design and modularization, mainly due to the fact that it provides a good solution for modularization. Some examples of those projects are SOPRANO (Wolf et al., 2008), which is based on the development of services that provide semantic contracts related to different levels of an elderly care scenario to specify its functionality and requirements to

other services. AMIGO (AMIGO, c, 2009), that make use of services to dynamically accommodate the different devices and technologies needed to autonomously operate in a home environment. The PERSONA project (PERSONA, 2009, a), on the other hand, distinguishes different types of components by using multiple event-based buses with subscriber and publisher components for direct input/output and context management, but relies on services for the abstraction of the high level functionalities.

The other approach for system modularization is the multi-agent paradigm. It is based on the division of the system into multiple autonomous components, called agents, that collaborate to resolve a functionality. The key characteristic is agent autonomy, as it implies that the agents must present other characteristics that make them very attractive to AmI [aitami][chen-rui], like loose coupling, reactivity, proactivity and robustness. To operate in an autonomous way, agents needs to react to changes in the environment, a good property for systems that will operate in such a highly dynamic environments as the real world. They also need to be proactive, because every one of them should pursue its own goals, taking initiative and interacting with others and the environment as needed, again a good characteristic for AmI systems that must adapt themselves to the environments and their inhabitants, learning from them in order to predict their needs. And finally agents must be robust to operate in an autonomous manner with very little human intervention. These characteristics have led to different AmI projects adopting the multi-agent paradigm as its technological base. Prominent examples are the work by Soldatos et. Al from the AIT (AIT, 2009) within the CHIL (CHILL, 2009) project (Soldatos et al., 2007), which presents a three tier architecture with a tier of agents for high level functionality that rely on the other two layers for sensing and signal processing, or the Intelligent Room (MIT Oxygen, 2009) project at MIT which uses agents through its Metaglue (Hanssens et al., 2002) and Hyperglue (Peters et al., 2003) systems to represent resources and their interactions. Chen Rui (Rui et al., 2009) from the Embedded Software and Systems Institute of the Beijin University of Technology present a general purpose AmI architecture with a multi-layer design and use multi-agent systems to implement it. In our own approach, the HI3 project, we have also proposed a multi-layer design (Paz et al., 2008, b), and we use multi-agent technology to implement the different elements that populate each layer.

Distributed operation

Highly related to the modularization problem and the nature of the environment in which AmI systems must operate, the components of an AmI system should operate in a distributed fashion. They need to rely on multiple heterogeneous devices to sense and act on the environment, and more importantly, they need to provide its functionality in an ubiquitous way, and thus need to operate multiple devices at multiple locations. This increases the importance of the communications capabilities that a middleware for AmI provides to its components.

With respect to the projects that use SOA, like SOPRANO or AMIGO, almost every one of them employs OSGi (OSGi, 2007), an specification for a dynamic module system for Java that is a de facto standard for SOA based systems. There exist numerous implementations of the OSGi platform specification that provide tools for the development, deployment and management of services. Due to the nature of SOA, systems using this approach employ RPC (Remote Procedure Call) like communications almost exclusively. That is, for instance, the case of SOPRANO and AMIGO. With respect to the technical solutions for RPC communications within OSGi, the initial versions of the OSGi specifications did not contemplate distributed operation, so numerous projects started to provide that functionality, the most widely distributed one is R-OSGi. The last revision of OSGi, release 4, version 4.1 (OSGi, 2007), contemplates support for the uPnP protocol and for the publication of services through the HTTP protocol. With respect to multi-agent based systems, they normally rely on ad-hoc message interchange for component communication and on FIPA (FIPA, 2009) compatibility for interoperability with other systems.

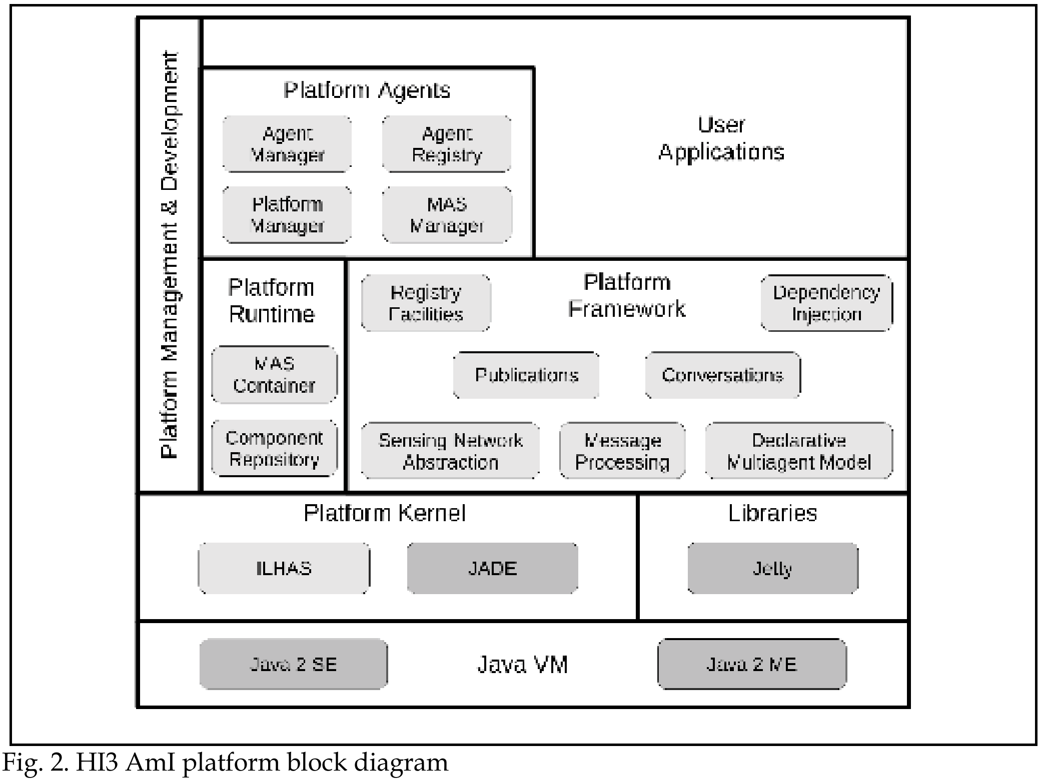

Fig. 2. HI3 AmI platform block diagram

We think that RPC like and simple message passing communications are not enough to support the complicated interactions between components that will be necessary to operate on an environment as complex and dynamic as the real world and to handle its relations with its inhabitants. Developers need more elaborate and high level communications tools to model and implement the interactions among its components. Event-based communications are a common solution for loosely coupled one-to-many communications, so they perfectly fit the requirements of AmI components. They have been used by several projects, like AMIGO, which considers an event based mechanism for context change management, PERSONA that introduces event buses as its primary communications, input/output and context management system. In HI3 we have also implemented an event based communications system for agents, providing them with the capability to publish information on publications that other agents can be subscribed to. Complex one-to-many, and many-to-many communications are also desirable in order to coordinate societies of components to pursue a common goal. This is also a typical problem of multi-agent systems, and as a consequence FIPA provides some facilities for conversations, especially for the case of one-to-many. In HI3 we have developed an extensible system to manage complex many-to-many agent conversations, every HI3 agent can initiate and participate in conversations whose behaviour is controlled by different conversational models that contain the logic to drive a particular type of conversation. This include how to invite new members into a conversation, how to leave a conversation or how to notify members of message reception.

System management

System management is often overlooked in AmI systems or at least, there is not much information about the techniques used by the different projects. Some of the systems are intended for home use thus, they try to avoid the need for management. However, when intended to be used in large installations like hotels or hospitals, where there will be professional staff supporting the systems, system management capabilities are a must. The general approach is to rely on the management capabilities of the underlying component technology, like remote service deployment on OSGi platforms. One noteworthy exception is the case of AMIGO and its OSGi Deployment Framework (AMIGO, a, 2009) that is able to dynamically deploy services taking into account their semantic descriptions. With respect to the HI3 project, we have developed a declarative model and its associated tools that make the declarative definition of multi-agent systems (and its components, agents, behaviours, etc.) possible. It uses a self designed URL system for component resolution, thus providing loosely coupled component aggregation. Currently it only resolves components that are available within its own component container, but we are working to support remote dynamic deployment and instantiation of components. Along with this system, HI3 provides an AmI system execution container that integrates a lightweight embedded web server to provide a remotely accessible web interface for component life cycle management and deployment. It also provides support for the components to use that web server to provide web interfaces for management or even for web service compatibility.

To illustrate a complete proposal for base middleware in an AmI system, intended to alleviate the problems of AmI system development, Figure 2 displays a block diagram of the AmI system development and execution platform developed for HI3. The platform uses Java as its programming language and, as mentioned before, is based on multi-agent technologies. It uses the JADE (JADE, 2009) agent platform as its base, and on top of that provides all the functionalities that have been describe throughout this section, such as component definition and description, high level communications capabilities like conversations and event-based communications, component management and searching capabilities and hardware abstraction facilities.

3.3 Context-awareness

In this section, we review various existing context-aware systems focusing on context middleware and architectures, which facilitate the design and development of context-aware services and applications. We discuss important aspects in context-aware systems and present some ideas related to context management in the HI3 architecture.

Many researchers have debated definitions of context for many years without reaching a clear consensus (Dourish, P., 2001). In the literature the term context-aware was first introduced by Schilit and Theimer (Schilit and Theimer, 1994) in 1994. They define context as "the location and identities of nearby people and objects and changes to those objects". Dey, in (Dey, 2001), reviews definitions of context and provides an operational definition of context as "any information that can be used to characterize a situation". This is the sense in which we understand the term context. The context defines the elements that must be observed in order to model the current state of the environment (situation). Additionally, context-aware applications must be able to dynamically adapt their behaviour to the current context without direct human intervention, in order to increase their usability and improve the end-user experience.

Dey and Abowd (Dey and Abowd, 2001) propose the division of context entities into three categories: places, people (individuals or groups) and things (physical objects). These entities can be described by attributes which can be classified into four categories: location, identity, status and time. Another popular way to classify context instances is the differentiation between the external and the internal dimension. The external dimension refers to context that can be sensed by hardware sensors, whereas the internal dimension is extracted from user interactions.

Related with the method used for context information acquisition, we can enumerate three different approaches (Chen et al., 2003):

- Direct sensor access. The applications gather the information directly from the sensors. This hinders system scalability, adaptability and distribution.
- Middleware infrastructure. This approach introduces a layered architecture that hides low-level sensing details. Hence, facilitates reusability and encapsulates the hardware dependent software.
- Context server. This approach extends the basic middleware based architecture by introducing distributed access to remote context data sources. We need to consider that probably the majority of devices used in ubiquitous environments are mobile devices operating over different network protocols.

Additionally, in order to manipulate and store context data in a computational form, we need to define a context model. There are different context modelling approaches that use different data structures for representing and exchanging contextual information: simple key-value models, object oriented models, logic based models or ontology based models. Ontologies are the most expressive models and are a very promising instrument for modelling contextual information (Strang & Linnhoff-Popien, 2004).

Focusing attention on AmI environments, there is a high level of consensus among researchers on the idea that it is essential to provide a framework for supporting context-awareness. One of the goals of a smart environment is that it supports and enhances the abilities of its occupants in executing tasks. In order to support the inhabitants, the smart environment must be able to detect the current state of the environment and determine what actions to take based on this context information. Therefore, applications in AmI environments need to be context-aware so that they can dynamically adapt themselves to different situations (Hung et al., 2003).

Related projects

Many ad-hoc context-aware systems designed to deal with predefined and restricted scenarios can be found in the literature. These systems may be optimized for this scenario but they aren't flexible or expandable. As a contrast, and following the global architectural approach we promote in this chapter, we will focus on middleware or architecture based systems. Middleware should provide uniform abstractions and independence from hardware. It must simplify the development and maintenance of context-aware systems. It should also simplify the interaction between different context-aware applications by means of the use of common abstractions and tools. In what follows we will introduce various frameworks that meet some requirements such as architectural approach, historical data management, context model definition, sensing abstraction model or security and privacy.

An example of agent based architecture for supporting context-aware computing in AmI environments is the Context Broker Architecture (CoBrA) (Chen, 2004). The core of this middleware platform is the intelligent context broker that manages a shared contextual model making it accessible to a society of software agents. The context broker also includes components that provide a context knowledge base, components that encapsulate the context data acquisition process and components that grant private access to the information managed by the system.

The Service-Oriented Context-Aware Middleware (SOCAM) project (Gu et al., 2004) defines an architecture for the rapid prototyping of context-aware mobile services. It includes a server that collects and manages context data using distributed context providers. Services can make use of different levels of context data in order to perform their context-aware tasks.

Another middleware platform based on a layered architecture is developed as part of the Hydrogen project (Hofer, 2002). The main particularity of this framework is that it manages a local and a remote context. The local context only contains information provided by local devices, the remote context provides the information that other remote devices know. The devices share their context information when they are able to establish a communications. With this strategy, a device's context consists of its own local context and a set of remote contexts obtained from other devices. The architecture also includes a software layer that abstracts the access to contextual information from the client applications.

The Gaia project (Roman, 2002; Gaia, 2009) defines another framework that extends typical operating system concepts to include context, location awareness, mobile computing devices and actuators. The project is focused on building applications in a generic way without assumptions about the current resources of an active space. The platform must be able to transparently locate appropriate devices, detect the addition of new devices to the system, and adapt content when data formats are not compatible.

The PRIMA project (PRIMA, 2009) is about machine perception of people and their activities. This project includes the development of a lightweight middleware intended to help developers to create ubiquitous applications. The authors argue that context is a key concept in ubiquitous computing and propose a layered architectural model for services based on human activity (Emonet, 2006; Crowley, 2006). This model defines a conceptual and operational division of the system in four layers: the sensor-actuation layer that encapsulates the diversity of sensors and actuators through which the system interacts with the world; the perception and action layer that operate at a higher level of abstraction than sensors and actuators; the situation model layer that allows the system to focus perceptual attention and computing resources according to the current state of the environment; and, at the highest level of abstraction, the service layer.

Other important projects in the area of AmI, such as AMIGO (AMIGO, c, 2009) or PERSONA (PERSONA, b, 2008), develop context-awareness frameworks as an important part of architectures that facilitates the development and integration of AmI applications. Both projects together with the PRIMA project share important design criteria, such as a layered approach with different levels of abstraction, use of ontologies to represent contextual information or the inclusion of components for user profiling and subscription/notification mechanisms.

Context model

We believe that an extensible, adaptable and efficient model for handling, sharing and storing context information is fundamental to providing effective and usable applications to a variety of users in scenarios with large variations in resources and activities. Context is too complex to be defined and programmed as a fixed set of variables. This demands the use of higher-level knowledge approaches together with reasoning and learning techniques.

The context model and context processing logic are the major components for providing intelligent, adaptable and proactive context-aware services and applications. As indicated before, ontologies provide an interesting formalism for specifying contextual information. Based on ontological models we can develop more sophisticated tools that provide a generic middleware platform to support context modelling in heterogeneous smart environments.

In the conceptual model of the HI3 architecture, we propose the division of AmI systems into four main layers (Figure 1). These layers represent different levels of abstraction and, as a consequence, the system can manage multiple views of the same contextual information in different layers and with different levels of abstraction. Hence, middleware must provide context aggregation capabilities that permit the composition of fine-grained context data in order to obtain higher-level context representations of the information. Middleware must also provide context interpretation capabilities to perform transformations of context data.

Awareness and notification

The awareness and notification capability must provide the functionality required to develop applications that can easily stay aware of any significant change in context. To achieve this, we propose a subscription/notification mechanism that allows services to register rules that specify what changes in context they should be notified of.

Historical information

Managing historical context data provides the possibility of implementing highly adaptable context-aware services. Furthermore, using learning algorithms, contextual information can be predicted to proactively adapt system behaviour to user habits. Related with historical data we can introduce in our architecture the concepts of granularity and period of validity. These properties are important to determine the relevance of contextual data to solve a certain task.

Information storage, privacy and security

We believe that it is necessary to use persistent storage systems to save coherent context data. This storage system must manage privacy policies and different levels of abstraction for data representation. Furthermore, is important that this component has facilities to query historical context data. Context usually includes sensitive information about people (location, activity, medical data, etc.). Hence, it is necessary to be able to protect privacy specifying policies and ownership of context information. We propose a transversal view of privacy and security components that operates in all the layers of the architecture. Thus, at every level of abstraction, one must incorporate mechanisms to support privacy, trust and security.

User modelling and profiling
A major requirement of AmI systems is that they must be capable of learning from the use and habits of the users in order to predict and anticipate them. Moreover, these systems must exhibit proactive, adaptive, predictive and autonomous behaviour. These goals are achieved by constructing, maintaining and exploiting user models and profiles, which are explicit representations of individual user preferences. Hence, an ambitious AmI architecture and context-aware framework must provide components and structures that facilitate the development of applications that make use of user models and profiles.

3.4 Mobility

Ambient Intelligence systems should provide an ubiquitous experience to its users, in other words, as far as possible, they should offer their services and functionalities independently of user location. For example, if the user is in his office accessing a travel planning application to prepare his next work trip and suddenly the user needs to go out to a client's office, the application should transfers its execution to his mobile phone, maybe by changing to simpler algorithms for data processing due to the lack of resources of mobile devices, and obviously changing the user interface. As soon as the user is in his car, the system transfers the functionality to the car computer in order to increase resources to use the more demanding algorithms and again changing the user interface, for instance to voice commands. Examples of this desired behaviour can be found almost for every functionality that an AmI system could provide to its users, except those that are highly related to a specific hardware device or installation.

Achieving these types of mobility capabilities is not an easy goal and is yet in an early research phase. To provide such capabilities to its developers, AmI architectures and their supporting middleware must support classical techniques from the fault tolerance field, such as component duplication and component state synchronization. They should also provide new techniques that make use of the semantic descriptions and contextual information in order to correctly move and duplicate functionalities between devices and spaces, taking into account component requirements, such as memory footprint or specific devices needed for its operation, as well as context information, such us preferences or even abilities of the users in order to select the correct user interface.

Mobility has been a hot topic in multi-agent systems research because with mobility agents gain numerous interesting capabilities (Oshima & Lange, 1999). For this reason, projects that use multi-agent paradigms have a huge advantage on this particular aspect. In fact multi-agent platforms like JADE provide basic support for agent mobility. Agent based approaches can focus on providing high level features that use their semantic and context systems to achieve intelligent component mobility.

With respect to SOA based approaches (usually based on OSGi platforms), OSGi does not support integrated mobility or fault tolerance features so, currently, a lot of efforts are being devoted to the development of OSGi extensions to support fault tolerance (Ahn et al., 2006; Filipe & Torr, 2009) that could be used as the basis for mobility features by reusing its component replication capabilities. Some work on SOA services mobility is starting to be reported. An interesting paper in this line is (Preuveneers & Berbers, 2008). It provides a good overview of mobility in pervasive systems and, in particular, it proposes solutions for mobility in OSGi based systems, also presenting an interesting implementation example.

3.5 Hardware abstraction

Ambient Intelligence systems are intended for operation in real environments, more specifically; real life social environments like homes, hospitals or hotels, inhabited by people, with the objective of making people's life easier and more comfortable. To achieve this, AmI systems must infer quality contextual information about the environment and its inhabitants. Complex algorithms and techniques are being proposed for this purpose but, in the end, all of them must rely on real data coming from the environment, and there is no other way to obtain those data than using sensing/actuation devices deployed in the environment or even on the people (wearable sensors, etc.).

There are many technologies competing for the domestic and social market, each one offering adequate solutions for its portion of the market. AmI software must be able to take advantage of these technologies but, the huge disparity of the different technologies and interfaces results in a series of problems:

- Difficulty of interoperating groups of devices of different natures.
- Portability of software components to different hardware platforms.
- Risk of translating the complexity of the hardware set-up to the software applications.
- Risks derived from the dependence on a given technology vendor.
- In real environments there may already be hardware elements present, such as classical domotics buses, which require their integration.

Solving these problems is not an easy goal, and the majority of AmI projects have mostly ignored them, adopting concrete technologies. This approach anchors the software solution to the selected hardware technologies, introducing unnecessary complexities on the software side and distracting developers from the core logic of its functionality. Furthermore, the software needs to be adapted to the hardware characteristics, loosing its generality and making its reuse on other project or environments difficult.

Almost all the projects working on integral architectural solutions for AmI have identified the hardware access as a key aspect. The good news is that all of them agree on the paradigm that should be applied to deal with this problem and are developing hardware abstraction layers that hide the complexities and differences of heterogeneous technologies. The main differences found between the approaches have to do with the design of those layers and the type of devices they are dealing with. On one hand there are projects that have focused on particular niches. For example, projects focused on home entertainment are mainly working on the interoperation of consumer electronic devices. This is the case of the INTERPLAY system (Messer et al., 2006), that tries to provide integration among home entertainment devices by using a three-layer architecture to support multiple device technologies and hide their heterogeneity.

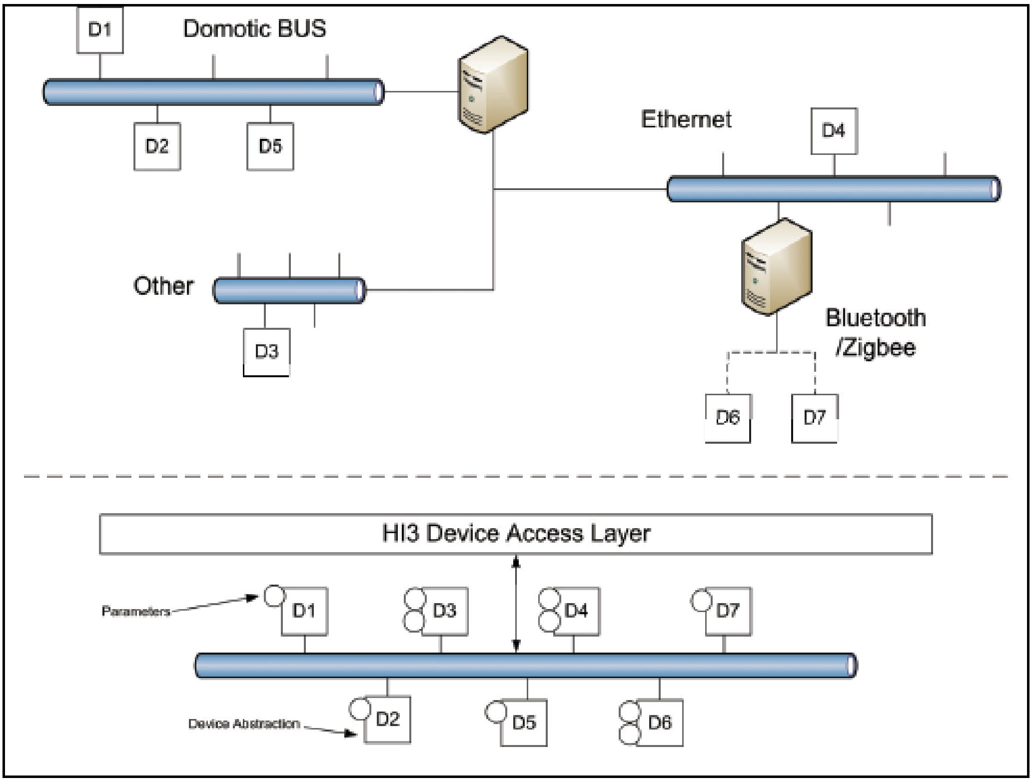

Fig. 3. Physical vs. Virtual hardware architecture

On the other hand there are projects that try to provide a general solution to the hardware integration problem. A prominent example is the AMIGO project (AMIGO, c, 2005) that has worked on the integration of heterogeneous technologies coming from domotics and consumer electronics. AMIGO relies on standardized solutions as much as possible, developing abstractions to hide the heterogeneity of the devices and translate their functionality to AMIGO compatible services. AMIGO proposes uPnP as a standard for hardware access so, when a non uPnP compatible device needs to bee used, AMIGO develops proxy services that use ad-hoc drivers to interact with the device and translate its interface. Proxies have been developed for two widely used domotic technologies, BDF and EIB. Chen Rui et al. (Rui et al., 2009) propose a similar solution to AMIGO, integrating technologies through the use of encapsulating proxies. The main difference is that they also provide a hardware solution to simplify the physical integration of technologies as well as the development of new devices.

From the beginning of HI3 we though that a solution for the integration of hardware technologies was the first step towards more reusable and maintainable AmI systems, so we started the development of the HI3 architecture from its hardware abstraction layers (Varela et al., 2007). Our approach is similar to the ones proposed by AMIGO and Chen Rui et al. It is based on the definition of a standardized paradigm for hardware access and the development of proxy-like devices to adapt current existing technologies.

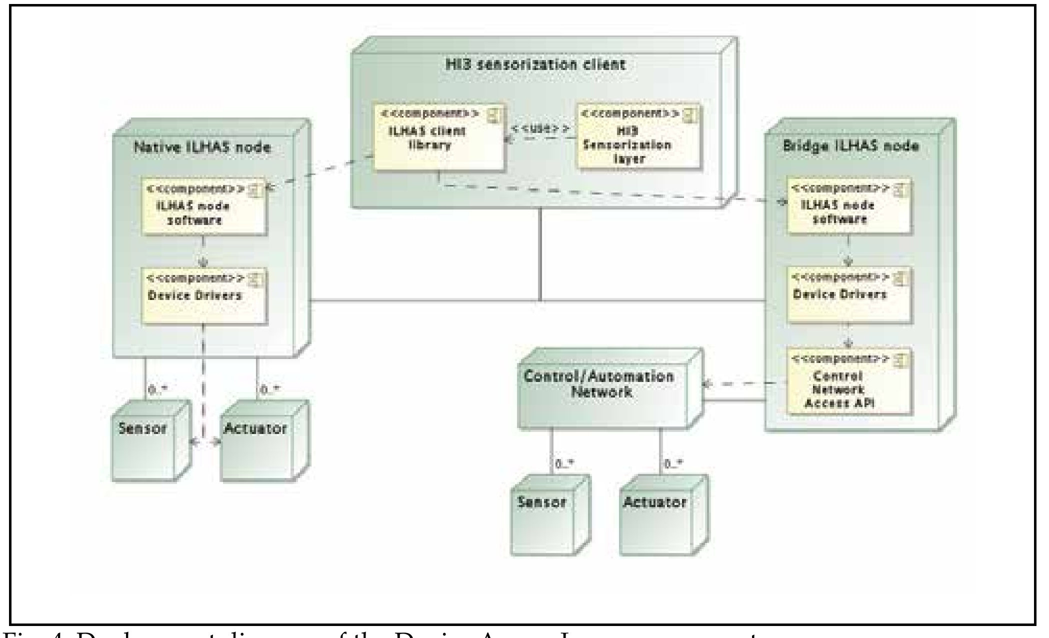

Fig. 4. Deployment diagram of the Device Access Layer components.

The HI3 project proposes a conceptual model based on a multilayer design with two layers in charge of abstracting hardware technologies. Going from the top down, the Sensing and Actuation layer is made up of agents that provide higher level views of hardware devices, for example aggregating them, applying some basic signal processing algorithms, etc. The Device Access Layer takes inspiration from the IEEE-1451 (NIST, 2009) standard (Varela et al., 2007). It is based on a distributed model inspired by the network and application layers of IEEE1451.1 standard and is supported by three basic pillars:

- Unified and generic view of the different hardware devices.
- Definition of a paradigm for the operating model of the sensing/actuation device network.
- Abstraction of the device network topology.

The first two abstractions provide functionality-based access to devices. They are accessed through read/write operations over parameters that represent the objects or states of the physical environment that a device can operate with. The third one is implemented as a network layer and a node abstraction that enables the integration of multiple and heterogeneous device networks.
This model is summarized in Figure 3 which displays the two visions of a complex scenario. On the top there is the physical vision of the installation, in which the software is exposed to the complexities of each different device. The bottom represents the homogeneous view the hardware abstraction subsystem provides.
The Device Access Layer is implemented through two components Figure 4, a software library to provide homogeneous and ubiquitous access to a network of devices (called ILHAS), and node software that can be used to integrate devices into that network. We have also developed a hardware prototype device (Varela et al., 2007) that integrates this node

software with COTS hardware, creating a good low cost solution for the development of new devices as well as hardware proxies for existing devices or technologies.

4. Applying HI3 technology

In the previous section of this chapter we have presented a review of different interesting approximations to provide Ambient Intelligence with models and supporting software to speed up the development of AmI systems. The HI3 architecture, a general purpose base architecture for AmI, has been presented along with these descriptions. In order to clearly illustrate how AmI support middleware, and more specifically the HI3 architecture, can help in the development of AmI software, we are going to show how it can be used to build a simple but interesting example that includes many of the typical requirements of AmI systems.

The example is a fairly typical one, a simple meeting management application. Similar examples have been used in numerous AmI articles to date as this type of application has functionalities that are easy to explain but presents complex underlying requirements like heterogeneous hardware integration, dynamic component collaboration, distributed operation or IA capabilities. It is an AmI application in charge of the management of one or more meeting rooms. It has to control all the devices involved in a meeting process, like the lights of the room, the projector display or the computer that shows the presentation. It also needs to autonomously know when a meeting can start by looking at who is in the room and checking if all the expected members have already arrived. In order to introduce some high level data processing techniques, we also want the application to autonomously extract user preferences about meeting room configuration.

In order to implement this example, we have at our laboratory a meeting room with all the required hardware devices:

- An EIB bus with light actuation devices, presence sensors and a collapsible projector display.
- A video projector with ethernet connectivity that is compatible with the PJLink protocol for remote control purposes.
- One computer with its video output connected to the video projector in order to display presentations.
- One computer with various bluetooth dongles that will provide a list of MAC addresses that are detected inside the room. Therefore, the users must simply carry a mobile device with bluetooth support in order to be located and identified.

Following the HI3 architecture model guidelines, those devices will be abstracted as ILHAS parameters that represents their device functionalities, like the state of the lights and operations to read/write it, the state of the presence sensor or a list of detected MAC addresses in the case of the bluetooth device that will be used to know who is in the room. The type and number of ILHAS parameters is standardized across devices that provide the same functionality, in other words, two light switching devices from different vendors, for example EIB and X10, will present the same parameters to the ILHAS subsystem. As the ILHAS library provides direct access to those parameters, homogeneous access to heterogeneous device technologies is effectively achieved.

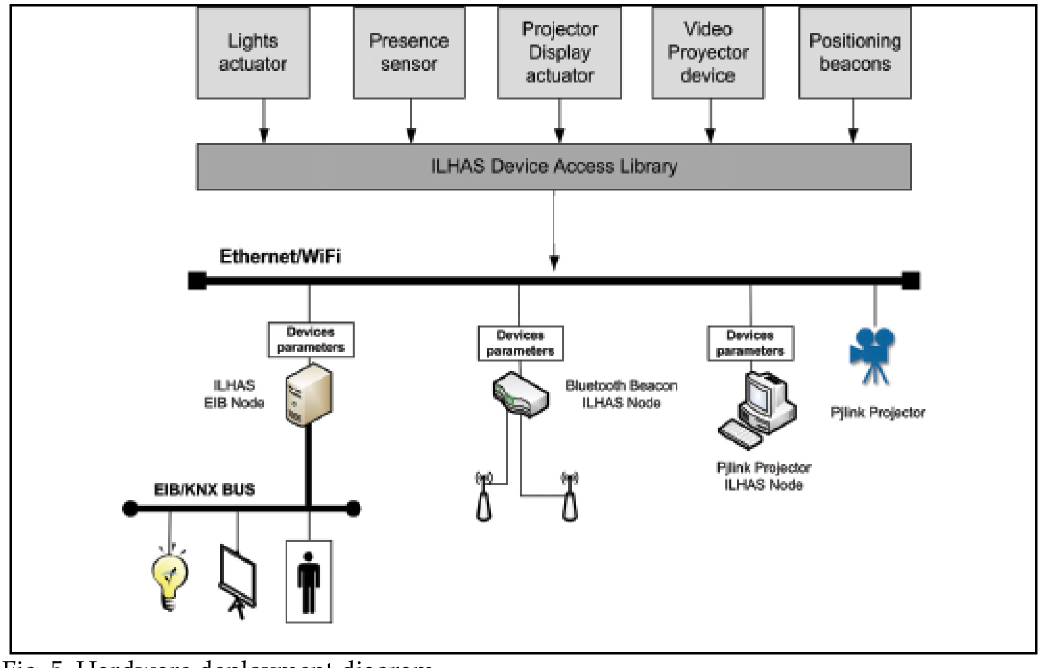

Fig. 5. Hardware deployment diagram

ILHAS provides also a homogeneous network topology on top of complex topologies that integrate multiple network technologies. As can be seen on Figure 5, there is at least one device adaptor for each technology that translates its concrete interaction protocols to the ILHAS parameter read/write paradigm. In order to use EIB devices, a bridge node has been implemented by using the ILHAS node software we talked about in the previous sections. It is deployed on a PC and connected to the EIB bus through a RS-232 interface using the Calimero EIBnet/IP tunneling library and the Tweety EIBnet/IP tunneling server from the TU Vienna (TU, 2009). With respect to the video projector, even though it has TCP/IP connectivity by using the PJLink protocol, a bridge node was also developed. This bridge node uses the PJLink protocol to interact with the projector and provides its functionality as parameters through ILHAS. This PJLink bridge node can be directly reused with other PJLink projectors. Finally, the bluetooth devices can be seen as a native ILHAS node because in this case there are no translations involved. The bluetooth dongles are deployed on a PC that runs an ILHAS node with device drivers that access those bluetooth devices.

As proposed by the HI3 architecture, all the devices are accessed through elements in the Sensing/Actuation layer that will translate them to a higher level view, by, for example, incorporating new information like their location, and providing automatic fault tolerance by transparently changing from one device to another in case of failure or by aggregating them. In this example we are going to keep this layer as simple as possible so it can be seen as redirecting proxies. It is important to notice that these elements are independent of the hardware technologies of the devices they represent because they access the hardware by using the ILHAS library, thus the video projector component can use any type of video projector, from a PJLink compatible one to an RS-232 device, as long as they are adapted to ILHAS.

As shown on Figure 6, the proxy components are elements of the Sensing/Actuation layer of the HI3 architecture. As such, they are implemented as multi-agent systems (in this particular case, they have only one agent) and they have an associated formal description of their functionality that is registered by the HI3 container at deployment time, and that can be used by other system elements to dynamically find them. The functionality descriptions of every element (in any of the layers) can be parameterized at deployment time or at runtime with specific values like, for example, a location in which the element operates.

Looking again at Figure 6, a high level view of the system's design is shown. As proposed by the HI3 architecture model we have designed the system as a set of loosely coupled components that collaborate to resolve the proposed functionality. Those components have been developed within each one of the proposed HI3 layers by looking at their purpose. Hardware access components are on the Sensing/Actuation and Device Access layers. General purpose components that can be reused by other components or projects are inside the Service layer. This is the case of components in charge of identifying the detected users of the system or controlling the set-up of a meeting room. Those components are examples of loosely coupled functionalities that, by having them modularized, can be directly reused in other applications and environments. Finally there are the components that provide concrete, not easily reusable functionalities; we call them applications, as they are the components that implement the use cases that a user expects from the system. In this example we have only one application, the meeting room management application. It relies on the available services to provide its required functionality and incorporates custom logic for its specific functions, such as, for example in this case, the profiling of the meeting room configuration.

We are now going to look more carefully at how the different elements resolve the proposed functionalities.

To achieve user location and identification functionalities a simple approach was applied. Users must carry a device with bluetooth support (for example a mobile phone) and we will use bluetooth dongles integrated in the system to detect those mobile devices. On top of these detection devices there will be a Presence service. It is associated to a location and knows who is in that location. It infers that information in a multi-modal way by using two types of information. Bluetooth devices to know who are in the room, and discrete presence sensors for fault tolerance and to alleviate the problems of the bluetooth technology like detecting people that is the contiguous room. Together with the Presence service there is the Identification service that uses the presence information and user data to generate user location and identification information. The communication between the Identification and Presence services, is achieved using the publish/subscribe capabilities provided by the HI3 middleware. The Identification service is subscribed to a Presence information publication. The same strategy is applied for the Presence service and the devices it uses. It is important to highlight that multiple services of each type can be present in the system at any time and the connection between them is performed dynamically. Their functionality is registered in the HI3 container and, as we have already indicated, can be parameterized, for example by the location they control. Thus, for example, when an identification service is deployed it will search for an adequate Presence service by using its standardized functionality description parameterized with its location.

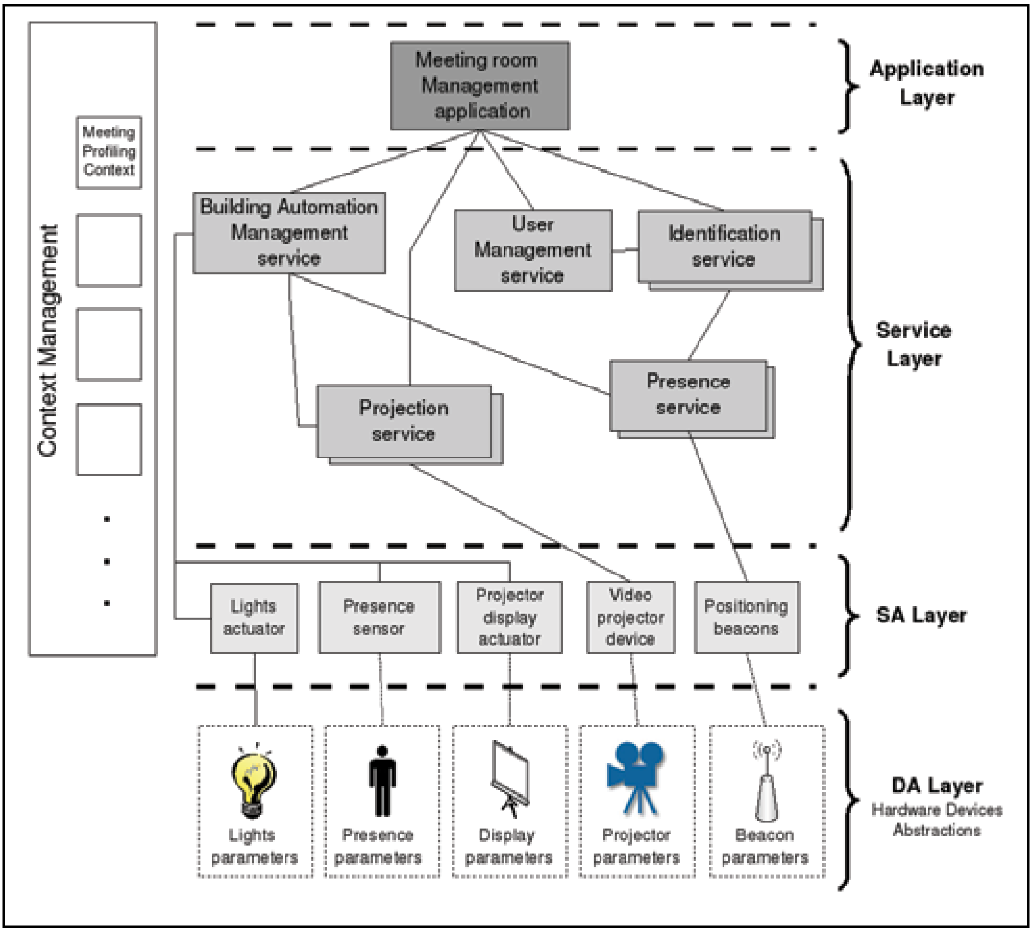

Fig. 6. Component diagram of the proposed example

The Projection service is similar in concept to the Presence service. It is in charge of aggregating and providing control over the typical devices found in a meeting room. There can be multiple meeting rooms in a building, so there is one Projection service for each room and at runtime it will search for partners (video projector, projector display, lighting, etc.) by using their descriptions parameterized with the location value. As with the case of the identification and presence services, this effectively achieves dynamic component composition and collaboration.

The meeting room management application uses all these services to control the devices involved in a meeting and to access information about the environment and its users, like the people present in the meeting, the state of the lights, etc. It applies profiling algorithms to this information in order to extract preferences about the lighting configuration that a particular user (or group of users) prefers for a meeting. More specifically, it monitors the state of the meeting room lights during meetings detecting patterns in their behaviour. For example, when the system does not know the preferences of a user it may, by default, turn off the lights when a presentation starts, if the user does not like that configuration it will turn on the lights (or a given subset of them) using any of the available actuators. This is the

information used by the system to learn the preferences of the user. This information is associated with the context and objective in order to increase the probability of the system automatically turning on some lights during a presentation when the context is similar.
In this simple example we have introduced typical AmI application requirements, like environment sensing and actuation through multiple paths, dynamic component composition and collaboration and even IA algorithm integration. It also has presented how an AmI general purpose architecture can speed up the development of AmI systems by providing solutions that are ready to use in multiple problems and a structured way of integrating them.

5. Conclusion

This chapter provides a review of some of the most important operational aspects of Ambient Intelligence as well as their bearing on the development of the software/hardware platforms that must support it. In this line, a series of issues that an architectonic approach to AmI software development should tackle have been identified and we have shown how the research community in general, and our laboratory, in particular, is dealing with them. Through the study and presentation of the HI3 architecture proposal and by relating it to other approaches found in the literature, the different solutions to the issues that arise and how they must be considered in the design and tool development phase have been visited. Some of these solutions have been taken to a high level of development through their use in other fields. This is the case of distributed operation technology, modularity and low level communication. Others are currently in an intermediate state, some good advances have been made, but much work is needed to fulfil requirements like ubiquity and full autonomous operation; this is case of technologies for issues such as security, mobility, fault tolerance or hardware access and configuration. Finally other areas are still in an early stage of development, some promising work has been carried out, but it is still isolated (not integrated) or assumes too many simplifications in order to deal with the problems. In this situation we find areas such as context-awareness, component self-organization and collaboration or natural human-computer interaction. As a consequence, it is easy to see that there is still much to do in this area and formal and structural approaches are still needed in order to achieve the operational degree that would be required of such a system operating in real life situations with humans. Obviously, it is a very interdisciplinary subject, that will require bringing together knowledge and technologies from very different fields, both technological and social, within a framework that really allows a systematic approach to the construction and adaptation of these structures without leading to a quagmire of ad hoc solutions or a complete deadlock as the systems scale. And it is also evident that to achieve this end powerful design and development tools and architectures will be a key issue.

6. References

Ahn, H., Oh, H., & Sung, C. O. (2006). Towards Reliable OSGi Framework and Applications, *Proceedings of the 2006 sympossiumon Applied computing*, Vol. 6, 1456-1461, Dijon, France, ACM, New York.

AIT. (2009). Athens Information Technology, Center of Execelence for Research and Graduate Education, http://www.ait.gr.

AMIGO (2009), a. AMIGO OSGi Deployment Framework, http://amigo.gforge.inria.fr/home/components/wp3/OSGi_Framework/index/index.html, 2009.

AMIGO (2009), b. AMIGO Project, http://www.hitech-projects.com/euprojects/amigo/.

AMIGO (2005), c. Specification of the Amigo Abstract Middleware Architecture.

Bernardin, K., Ekenel, H. K., & Stiefelhagen, R. (2007). Multimodal identity tracking in a smart room. *Personal and Ubiquitous Computing*, Vol. 13(1), Issue 5, 25-31.

Brdiczka, O., Crowley, J. L., & Reignier, P. (2009). Learning situation models in a smart home. *IEEE transactions on systems, man, and cybernetics. Part B*, Vol. 39(1), 56-63.

Chen, H. (2004). An Intelligent Broker Architecture for Pervasive Context-Aware Systems, PhD Thesis, University of Maryland, Baltimore County.

Chen, H., Finin, T. and Joshi, A. (2003). An ontology for context-aware pervasive computing environments, *The Knowledge Engineering Review*, Cambridge University Press, Vol. 18, pp.197–207. 2003.

CHIL (2009). CHIL - Computers In the Human Interaction Loop, http://chil.server.de.

Crowley, J., Brdiczka, O., & Reignier, P. (2006). Learning situation models for understanding activity, *International Conference on Development and Learning*.

Dey, A. (2001). Understanding and using context, *Personal and ubiquitous computing*, 5(1), 4–7. Springer.

Dey, A. K. and Abowd, G.D. (2001). A conceptual framework and a toolkit for supporting rapid prototyping of context-aware applications, *Human-Computer Interactions (HCI) Journal*, Vol. 16, Nos. 2–4, pp.7–166. 2001

Dourish, P. (2001). Where the Action Is: The Foundation of Embodied Interaction, MIT Press, Cambridge, 2001.z

EMBASSI (2009). EMBASSI - Multimodal Assistance for Infotainment and Service Infrastructures, http://www.embassi.de.

Emonet, R., Vaufreydaz, D., Reignier, P., & Letessier, J. (2003). O3MiSCID, a Middleware for Pervasive Environments, *1st IEEE International Workshop on Services Integration in Pervasive Environments*.

Filipe, C., & Torr, L. (2009). Fault Tolerance in the OSGi Service Platform.

FIPA (2009). FIPA - Foundation for Intelligent Physical Agents, http://www.fipa.org.

Gaia (2009). Gaia Project, http://gaia.cs.uiuc.edu/

Gu, T., Pung, H.K. and Zhang, D.Q. (2004). A middleware for building context-aware mobile services, *Proceedings of IEEE Vehicular Technology Conference (VTC)*, Milan, Italy.

Hanssens, N., Kulkarni, A., Tuchinda, R., & Horton, T. (2002). Building agent-based intelligent workspaces, *ABA Conference Proceedings* (p. 675–681).

Hofer, T., Schwinger, W., Pichler, M., Leonhartsberger, G. and Altmann, J. (2002). Context-awareness on mobile devices - the hydrogen approach, *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, pp.292–302.

Hung, N., Ngoc, N., Hung, L., Lei, S., & Lee, S. (2003). A Survey on Middleware for Context-Awareness in Ubiquitous Computing Environments, Korean Information Processing Society Review, 1-20.

JADE (2009). JADE - Java Agent DEvelopment Framework, http://jade.tilab.com/.

Klein, M., Onig-ries, B. K., Ussig, M. M. (2005). What is needed for semantic service descriptions - a proposal for suitable language constructs, *International Journal of Web and Grid Services*, 1(3/4), 328-364.

Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M., Shafer, S., et al. (2000). Multi-camera multi-person tracking for easyliving, *IEEE Workshop on Visual Surveillance* (Vol. 6, pp. 1-8).

Messer, A., Song, H., Kumar, P., Kunjithapatham, A., & Sheshagiri, M. (2006). Interplay: a middleware for integration of devices, services and contents in the home networking environment, *3rd IEEE Consumer Communications and Networking Conference*, pp. 1083-1087.

MIT Oxygen (2009). MIT Oxygen Project – E21: Intelligent Spaces, http://oxygen.lcs.mit.edu/E21.html.

NIST (2009). NIST IEEE-P1451 Draft Standard, IEEE1451, http://ieee1451.nist.gov/.

OSGi Alliance (2007). OSGi Service Platform Release 4 Version 4.1, http://www.osgi.org/Download/Release4V41.

Oshima, M., & Lange, D. B. (1999). Seven Good Reasons For Mobile Agents, *Communications of the ACM*, 42(3), 88-89.

Paz, A., Varela, G., Monroy, J., Vazquez, S., & Duro, R. J. (2008), a. HI3 Project: Software Architecture System for Elderly Care in a Retirement Home, *3rd Symposioum of Ubiquitous Computing and Ambient Intelligence*, 11-20, Salamanca, Spain.

Paz, A., Varela, G., Monroy, J., Vazquez, S., & Duro, R. J. (2008), b. HI3 Project: General Purpose Ambient Intelligence Architecture, *Proceedings of the 3rd Workshop on Artificial Intelligence Techniques for ambient Intelligence*, AITAmI08, 77-81, Patras, Greece.

PERSONA (2009), a. PERSONA Project, http://www.aal-persona.org.

PERSONA (2007), b. Full version of ambient assisted architecture specification, http://www.aal-persona.org/deliverables/D3.1.1-Full_version_of_ambient_assisted_architecture_specification.pdf.

Peters, S., Look, G., Quigley, K., Shrobe, H., & Gajos, K. (2003). Hyperglue: Designing high-level agent communication for distributed applications.

Preuveneers, D., & Berbers, Y. (2008). Pervasive services on the move: Smart service diffusion on the OSGi framework, *Lecture Notes in Computer Science*, 5061, 46–60. Springer.

PRIMA (2009). PRIMA Project. http://www-prima.imag.fr

Roman, M., Hess, C., Cerqueira, R., Ranganathan, A., Campbell, R.H. and Nahrstedt, K. (2002). A middleware infrastructure for active spaces, *IEEE Pervasive Computing*,Vol. 1, No. 4, pp.74–83.

Rui, C., Yi-bin, H., Zhang-qin, H., & Jian, H. (2009). Modeling the Ambient Intelligence Application System: Concept, Software, Data, and Network, *IEEE transactions on systems, man, and cybernetics. Part C.*, 39(3), 299-314.

Schilit, B. and Theimer, M. (1994). Disseminating active map information to mobile hosts, IEEE Network, Vol 8 pp 22-32, 1994.

Soldatos, J., Dimakis, N., Stamatis, K., & Polymenakos, L. (2007). A breadboard architecture for pervasive context-aware services in smart spaces: middleware components and prototype applications, *Personal and Ubiquitous Computing* , Vol 11, Issue 3, 193-212.

SOPRANO (2007). Review state-of-the-art and market analysis, http://www.soprano-ip.org/ecportal.asp?id=350&nt=18&lang=1.

Strang, T. and Linnhoff-Popien, C. (2004). A Context Modeling Survey, *First International Workshop on Advanced Context Modeling, Reasoning and Management, UbiComp*.

Tapia, E., Intille, S., & Larson, K. (2004). Activity recognition in the home using simple and ubiquitous sensors, *Lecture Notes in Computer Science*, 158–175. Springer

TU Vienna (2009). Institute of Computer Aided Automation, Automation Systems Group, https://www.auto.tuwien.ac.at/a-lab/knx-eib.html.

Vallee, M., Ramparany, F., & Vercouter, L. (2005). Dynamic service composition in ambient intelligence environments: a multi-agent approach, *Proceeding of the First European Young Researcher Workshop on Service-Oriented Computing* (pp. 1-6).

Varela, G., Paz-Lopez, a., Vazquez-Rodriguez, S., & Duro, R. J. (2007). HI3 Project: Design and Implementation of the Lower Level Layers, *2007 IEEE Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems,* 36-41, Ostuni, Italy.

Wolf, P., Schmidt, A., & Klein, M. (2008). SOPRANO – An extensible, open AAL platform for elderly people based on semantical contracts, *Proceedings of the 3rd Workshop on Artificial Intelligence Techniques for ambient Intelligence*, AITAmI08, 11-15, Patras, Greece.

2

Services Everywhere: an Object-Oriented Distributed Platform to Support Pervasive Access to HW and SW Objects in Ambient Intelligence Environments[1]

Jesús Barba, Félix Jesús Villanueva, David Villa, Francisco Moya, Fernando Rincón, Maria José Santofimia and Juan Carlos López
University of Castilla-La Mancha
Spain

1. Introduction

The Ubiquitous Computing concept was first defined by Mark Weiser (Weiser, 1995) and it refers to a new computing era where electronic devices merge with the background. People make use of those electronic devices unconsciously, focusing just on their needs and not in how to accomplish them.

The concept of Ambient Intelligence (Ducatel et al., 2001), lying on the ubiquitous computing paradigm, refers to those environments where people are surrounded by all kind of intelligent intuitive devices, capable of recognising and responding to their changing needs. People perceive the surroundings as a service provider that satisfies their needs or inquiries in a seamless, unobstrusive and invisible way.

It is generally agreed that AmI (Ambient Intelligence) will have a great impact in economy and society. The potential of AmI technologies in various application areas has been object of numerous studies. For example, the *IPTS/ESTO Ambient Intelligence in Everyday Life Roadmap* (Friedewald & Da Costa, 2003) report analyzes key application areas in order to find out which are the technological requirements that must support the functions that will make the difference in each of those application areas (housing, mobility and transport, shopping and commerce, education and learning, health, culture, leisure and entertainment).

But AmI is not only its technological facet. There is also a social and a politics dimension besides the devices and software upon which the intelligent environments are built. Technology must be helpfull, work with users and not against them trying to pull down the wall of the natual resistance of the human being to revolutionary changes.

---

[1] This work has been partially funded by the Spanish Ministry of Science and Innovation under the project DAMA (TEC2008-06553/TEC), the Regional Government of Castilla-La Mancha under project RGRID (PAI08-0234-8083) and the Spanish Ministry of Industry under project CENIT Hesperia.

Putting the focus on the technological dimmension, the requirements for AmI to become real are varied and can be classified in two domains or computational areas (ISTAG, 2003): technologies for *ambience* and *intelligence*. An intensive reasearch in each one of them, by its own, is not a guarantee of success but an additional effort in finding the mechanisms to ensure the integration of components and devices in AmI systems must be done. Moreover, such integration must be perfomed in a seamless way.

Therefore, the importance that the *technologies for integration* play in AmI is out of any doubt. We strongly believe that the different research challenges in this area are in the core of the success of AmI. However, currently, many of the AmI demands for a hardware platform and software architecture, suitable for its needs, have not got a satisfactory response yet. The ISTAG in 2003 identified the following features that a platform for AmI must include: *abstraction, automatic composition, interaction management, computational efficiency, creativity, scalability and evolution* and *dependability*.

It can be easily checked that we are sill far from such vision since AmI environments are currently limited to a few demonstrators in the research centres, and the incorporation of the AmI technology to real scenarios and applications is taking place extremely slowly. Next, we try to be more precise and analyze in deep, from our point of view, the main reasons of such delay in the realization of AmI.

Within the layered vision of an AmI system, the lowest level is in charge of collecting data that will be translated into contextual information by upper levels. This layer is mainly composed by sensors, small devices with limited capabilities (memory and computational power). At this level, it can also be found a set of actuators that execute the commands coming from the layers that generate decisions from the data gathered by the sensors. On top of the *sensors & actuators layer*, a set of nodes with sufficient computing capabilities interpret the raw data, extracting the knowledge contained in them (in an automatic way). Moreover, the nodes of the upper levels may provide value-added services like the collection of statistical data or the integration with business processes.

So, in the basis of any AmI application there is a *complex, extremely large distributed system* containing not only nodes with uneven features regarding execution speed, battery life or interfaces but also a *communication infrastructure* that surrounds them and probably combines several *communication protocols* (i.e. RFID, Wi-Fi, WiMax, Bluetooh, etc). Such mixture of properties conform a *heterogeneous platform* that is intended to support the delivery of value added services to software components that are built on top of it.

It is difficult to tackle with heterogeneity since it poses additional difficulties to the development of AmI platforms and applications with the characteristics previously mentioned. First of all, programmers should be aware of such heterogeneity, inherent in AmI systems, and deal with several languages, programming interfaces, network protocols, etc. The time invested by the developers in these affairs is time they are not using to implement the *intelligence* of the nodes that, in any case, is what makes the difference. Also, the productivity of programmers will increase since they do not have to worry about the implementation details of the integration and communication layers.

To sum up, it is ***mandatory in AmI environments a tailored middleware platform to help the application and service developers with the integration of components in a simplified and seamless way***. Such middleware platform must not only provide the necessary semantic to connect and make the different elements in the system interoperable (acting as the glue between them) but also must offer a set of tools, methods and even embedded services (i.e. composition and reconfiguration services) to enable the rapid exploration and prototyping of new systems or more complex services.

Although AmI environments have much in common with distributed objects platforms developed in the nineties (i.e. CORBA, EJB, DCOM …), and also with current grid computing platforms (i.e. Globus, gLite …), such distributed platforms where not originally conceived thinking of the special needs of AmI. Therefore any attempt to extend standard middlewares and apply them to specific AmI applications or scenarios must be carried out carefully, trying not to inherit the deficiencies of the original middleware concerning AmI. On the other hand, such adaptation process represents an extra effort in order to make the former middleware converge with the unforeseen requirements imposed by the new scenario. Also, the development of new services and communication means for the standard middleware is not for free and, probably, their reuse in other AmI applications can be hard to accomplish.

To overcome the limitations of current approaches, the ARCO research group at the University of Castilla-La Mancha[2] started to work on a global solution, specifically conceived for AmI environments, based on its wide experience in distributed object platforms and application oriented middlewares[3].

Our proposal leverages the achievements of standard middlewares in regular networking distributed platforms and comprises:

- A standard communication platform which allows developers to integrate heterogeneous devices and networks.
- A common development framework to build new services or easily integrate legacy ones.
- A set of services that provide the developers with advanced features such as service discovery, positioning, management of reconfigurable hardware, migration, event channels, etc.

Throughout the rest of this section we sketch the main features and advantages of our work and how it effectively answers many of the major challenges in platform and software design for AmI as stated by the ISTAG.

1.1 A holistic approach to the major AmI challenges

The main objective of the Object-Oriented distributed Platform for AmI (from now on OOPAmI) is to address the demands of the emerging AmI technologies from an *integrated point of view, based on the Object-Oriented distributed paradigm*. Next, we detail how this primary goal is achieved:

- *Abstraction.* Both services and devices are modelled as objects. The concept of object *effectively abstracts* the hardware and network technology used to deployed the services in AmI applications. This allows service developers to concentrate their effort in achieving better and more intelligent algorithms instead of worrying about communication and platform dependent problems.

---

[2] http://arco.esi.uclm.es

[3] Among the projects handled by the ARCO research group covering these topics it is worthy mentioning those regarding domotic, homeland security and ubiquitous & pervasive computing.

- *Automatic composition*. On top of the object communication engine, a set of *Intelligent Agents* implement a reasoning engine which main responsability consists in responding to incomming events, needs or requirements by means of service composition. The so called, *Multi-Agent Service Composer System* is implemented using the Jadex platform.
- *Computational Efficiency*. OOPAmI is able to integrate in the platform architecture: (1) low footprint devices such as eight bit microcontrollers (micro-components); (2) full servers or mobile devices (i.e. PDAs, SmartPhones) with enough power of computation to run complex tasks or algorithms; and (3) application specific hardware in form of reconfigurable logic or Systems-on-Chip. The variety of the devices that OOPAmI can handle makes our platform flexible and easy to adapt, depending on the level of performance demanded by the target application. The transparent integration of fully customized hardware devices allows OOPAmI to cope with the most complex tasks. Besides this, the use of low cost, small computing devices for the simplest services (i.e. retrieving data from sensors or the control of the actuators) helps to achieve the maximum resource usage efficiency, lowering the final cost of the system.
- *Scalability and Evolution*. OOPAmI defines and implements a migration service. Such migration can take place from: (1) software nodes to software nodes (with different computing capabilities); (2) software nodes to hardware nodes and vice versa. The migration service facilitates the management of a punctual increment of the work load in the system, enabling the use of the whole available resources in the system. Once again, a set of *Intelligent Agents* are in charge of reasoning the better action in each scenario in order to provide the desired degree of quality of service. The *Multi-Agent Migration Service* uses the above mentioned migration service and a reconfiguration service to deploy the bitstreams on the reconfigurable logic devices where necessary. New hardware designs and algorithms can be incorporate to the system anytime.
- *Dependability*. Run-time failure management and security are also considered in OOPAmI. Transient and permanent errors in electronic devices are handled using different approaches: (1) utilization of replicas; (2) replacement of the affected node (i.e. downloading a new bitstream that fixes the malfunction); and (3) movement of the objects that run on the broken node. In OOPAmI these three scenarios are transparent to the developers due to: (a) a special distributed location service with extended capabilities; (b) a failure detection and notification service; and (c) a persistence service which is used by the objects to periodically save its internal state in order to be retrieved latter in case of failure.
- *Software and Service Architecture*. OOPAmI makes it use of the *Distributed Object Based Services* (DOBS)[4] that comprises several research projects within the ARCO group. The first implementation of DOBS targeted home services and, after several iterations it has evolved into a development and integration framework for AmI services. The Multi-Agent Service Composer System relies on DOBS and a OWL ontology to perform automatic composition of services.

[4] http://arco.esi.uclm.es/dobs

- *Design*. A unified design flow for DOBS objects is proposed. DOBS objects can be implemented on Wireless Sensor Network devices, mobile devices, computers or hardware.
- *Integration*. From an integration point of view, objects unify the way services are accessed which leads to better and low effort integration mechanisms.
- *Business model*. Nowadays, the role of different companies in the digital home market (the prelude of real AmI environments) is reduced to the engineering of costly projects regarding the control of *smart building*. Such projects are mainly based on the concept of *residential gateway* which is totally opposite to the openness principle demanded by AmI. It is important for AmI projection to open the market to new actors establishing open architectures and protocols. Roles as "service providers" or "resource providers" may then be possible so that a competitive market can be established. In this line, OOPAmI is a unified middleware *able to interact with standard middlewares* (i.e. CORBA, Internet Communication Engine) which facilitates the integration of new services or devices from any vendor.

To help the reader to visualize and understand the numerous contributions of OOPAmI, Table 1 summarizes its most relevant features and their relation with the requirements for the best AmI platform.

OOPAmI platform features and components	**Application to AmI environments**
• "Object" as the modelling concept	• Abstracts the platform • System-level focus • Helps to cope with heterogeneity • Offers the necessary semantics to integrate services and devices • Reusability and interoperability
• Multi-Agent Service Composer System	• Automatic composition of services
• Reconfigurable nodes	• Easy management of adaptive systems • Migration • Performance and flexibility
• Event channels	• Platform-independent data communication and acquisition
• Distributed Object Oriented programming model	• Single programming model • Easy development of algorithms and applications • Homogeneous view of the system • Automatic generation of embedded code

OOPAmI platform features and components	Application to AmI environments
• Low cost nodes (microcontrollers)	• Flexibility • Maximum resource usage efficency • Computational efficiency
• Multi-Agent Migration Service	• Scalability, evolution and dependability
• Distributed Object Based Services	• Automatic composition of services • Easy devlopment of services
• Unified design flow	• Design of services, hardware and embedded software
• Interaction with standard middlewares	• Openness

Table 1. Main OOPAmI features and their application to AmI platforms.

2. Previous work in middlewares for AmI

Due to the huge number of knowledge fields that AmI includes, there are many previous works related with AmI hot topics (integrated circuits, artificial intelligence, etc.). However, in this section we will only focus on those works regarding the middleware technology used to overcome some of the problems in AmI environments.

The middleware has an increasing number of tasks associated in AmI environment, and its design has to be carefully delimited to cope with all of them. Between these tasks we can mention homogeneous access to AmI devices and resources, with independence of the location, operating system, technology used, language used in the implementation, etc. of the services.

Maybe one of the key points of the approach is the interoperability between different technologies and services. This problem is not resolved yet. In the meanwhile some technologies are called to play an important role in the AmI environments, and therefore their integration must be considered:

- Radio Frequency Identification (RFID) starts to be a predominant technology on the identification field and all related applications (e.g. vehicles fleet control, goods tracing). RFID is also associated with applications that require basic information storage (i.e. identification cards). RFID is becoming indispensable in applications for user interaction. Usually, the RFID tag has scarce resources and existing middlewares can only be supported by the reader (generally attached to a desktop computer).
- Wireless Sensor/Actuator Networking (WSANS) is also another key technology in the future of AmI environments. AmI requirements of monitorization and control of the real environment makes use of WSANs as the interface to the real world.
- Body embedded devices. Currently, there is significant research in devices attached to clothes, or even directly to the human body. Going beyond of body area network, these devices supply data to the environment such as people health

indicators, preferences, and even emotional state that could hardly be extracted with any other technologies (for example, video analysis).
- Vehicular Ad-hoc Networks (VANET) is another emerging field to integrate in the AmI environment since the car constitutes our main mean of transport work and it is part of our daily live.

Some approaches that try to solve the problem of heterogeneity have chosen the java language (for being a multiplatform language) (Sacchetti et al, 2005) or XML and Web Services for the same reason (Issarny et al, 2005), (Perumal et al, 2008), etc. However, web services impose stronger requirements to the devices because they need to support a set of protocols that some of the previously mentioned technologies are unable to support. In the same way, Java requires its java virtual machine that, even in its more reduced version, is too big for devices like wireless sensor networks devices, body embedded devices, etc.

Once again, we must highlight that all the technologies mentioned before (but VANET) encompass devices with scare resources. This is why seamless integration of these technologies requires a light middleware (see section 3.1).

On the other hand, a middleware for AmI environments must be enhanced with extra responsibilities. It must provide advanced services and features to the developers, such as the ones listed below:

- People and resource positioning services: several AmI scenarios require knowing people location, in order to adapt its functionality.
- Automatic service composition: in a real AmI environment, the middleware should compose services that are hard to predict.
- Dynamic discovery of new devices and services: An AmI environment should be dynamically built upon the spontaneous appearance of devices and services., in order to put them all to work together.
- Developer assistance. For a better acceptance of any middleware technology, a simplified design flow based on a complete toolchain is recommended. One example of a very important facility a middleware should provide is a simulation platform. This simulation platform should emulate the devices, services, people and the information flow of a real AmI environment. There are some works with this idea in its aim, like for example (Maly et al., 2008), but usually they are not associated to any middleware so the developers can only emulate standalone algorithm with no interaction with emulated devices and services.

Most of the middlewares developed for AmI environments have not considered these and other aspects (as those enumerated in section 1.1).

The IST Amigo (Sacchetti et al., 2005) project has developed a Java middleware based on the OSGi platform (OSGi Alliance, 2006), extending the concept of residential gateway[5]. IST Amigo middleware also supports the development of services using HTTP/SOAP remote procedure calls. In our opinion, the role of a residential gateway has been encouraged by telecommunication companies with the intention of promoting the client loyalty and extending the service offer. The residential gateway approach has been shifted due the success of other devices that can also play the role of residential gateway (i.e. mobile phones, TV, etc.). The vision of a centralized point for service has definitely changed.

---

[5]A residential gateway is a single device that interconnects all networks (external and internal) and devices in the environment.

As we argued before, supporting the Java Virtual Machine associated to JAVA, or the protocol stack for HTTP/SOAP, imposes strong requirements to the devices that can be integrated using a middleware. The middleware developed in the IST Ozone project (Gelissen, 2005) is also based on web services developed in JAVA, sharing the same problems than Amigo Middleware.

These web services-based middlewares use WSDL (*Web Service Description Language*), an XML based language for service description. Nevertheless, although XML was designed with human readability in mind, a non trivial service description is hard to understand and use from a developer point of view. Service descriptions should be a natural action for service developers, not a maze of syntactical artifacts (as it happens with WSDL). With this principle in mind, our vision about service description simplifies WSDL based approaches because: (1) we use a simpler language (IDL) for service description; (2) we decouple the description of the interface from its attributes, simplifying the definition of complex devices; (3) we provide tools for attribute modelling.

Agent oriented middlewares constitute another approach for AmI. The works presented in (Wu, 2008) and (Marsa, 2006) are an exponent of this approach. In this type of works, the authors generally try to model human behaviour, or apply artificial intelligence techniques to AmI environments (Ramos et al, 2008) by means of software components called agents (they are typically programmed using the Java language for agent implementation due its possibility of serialization).

We have performed a short travel around what is done in the state-of the art in middlewares for AmI. Many of them only propose partial solutions to some facets of the problem as service composition, network interoperability or service discovery. Others do not even consider important features demanded in AmI middlewares (integration of small devices). We are looking for an integral approach for AmI development as we will describe in the following sections.

3. Integration of heterogeneous devices in OOPAmI

In this section we are going to describe how OOPAmI addresses the heterogeneity problem in AmI environments. One of the most remarkable features of our platform is the capability of integrating devices of different nature based on the *distribute object paradigm*. As justified in the introduction of this chapter, the concept of object provides the necessary semantic in AmI environments to get interoperable heterogeneous nodes due to its capability to abstract the implementation details of the network.

This approach is already present in many standard distributed object middlewares and it has been proved to be useful in the design of new and challenging applications for ubiquitous computing and ambient intelligence environments.

Nonetheless, embedding standard object middlewares require too much computing resources in the target devices in order to implement the whole middleware protocol features. The strict requirements imposed by this approach limit the kind of devices that can be incorporated to the AmI network. For example, the facto obligation of an operating system (i.e. to provide the access to the network interface) knocks out small devices with enough capabilities to implement the object or service functionality. Current solutions do not concern this matter and propose the utilization of oversized devices for the

implementation of simple nodes where most of the resources are assigned to the execution of OS routines or unnecessary middleware services.
Conversely, OOPAmI allows the incorporation of low cost devices to the platform[6], removing the need of the OS and the middleware burden in the nodes without sacrificing interoperability.
The principle behind this revolutionary approach can be summarized in the following sentence: *"Although it is important that each device looks like a distributed object, it is not essential that they are actual distributed objects. If devices are able to generate coherent replies when they receive redefined request messages then the system will work as expected"*. (Villanueva et al. 2007).
Such principle is also applied in OOPAmI to another special kind of nodes called *hardware objects*. A hardware object (HwO) is a custom integrated circuit that will perform a complex task or implement a service.
The shift in the utilization paradigm of dedicated hardware in OOPAmI, is considering such custom devices as *autonomous* entities in the network. Traditional approaches, view dedicated hardware as slave accelerator units under the supervision of a master controller (typically a microprocessor). Once again, the complexity of the infrastructure surrounding the special hardware unit is almost entirely given to run the OS and middleware services to provide interoperability. Thus, the cost of the supporting platform rises and so does the cost of programming the embedded software.
To avoid the dependence of custom hardware in a processor-based computation node, we have ported the kernel of a communication middleware onto a pure hardware implementation. The resulting communication infrastructure eases the integration of HwOs within the AmI network since they are able to understand the protocol messages of the underlying middleware.
The remainder of this section is dedicated to explain the architecture, tools and methods that enable small devices and custom hardware nodes to be incorporated in AmI application in a seamless way. Thus, the resulting middleware platform that sustains the services above the integration level is endowed with unforeseen features and devices in other commercial solutions. To exemplify the concepts, methods, tools and prototypes developed to this end we have chosen ZeroC ICE (Henning & Spruiell, 2008), an excellent CORBA like middleware, but the same approach is also applicable to other middlewares.

3.1 Ultra low-cost nodes in AmI platforms

When a node in the AmI network just holds an application-specific server (i.e. read a magnitude value from a sensor), the service, the whole communication engine and its API can be handled by a special object implementation called *picoObject*. A picoObject lacks in a local communication engine and it does not need of object adapters, marshalling routines, etc. We just need to implement the message handling code for the middleware protocol messages whose destination is an object placed at the device.
Nonetheless, for the rest of the network a picoObject behaves as a usual object. It provides a network level interface without significant differences with respect to a standard object. In our case, picoObjects are fully compliant with *ICE protocol*, the network level contract in ZeroC ICE.

---

[6] Since miniaturisation is a must to accomplish with the *very unobtrusive hardware* principle, is mandatory to offer an integration means for low cost devices with a minimal footprint.

PicoObjects can also handle client-side communications using similar techniques. Client-side messages are composed as a set of templates with just the bare minimum configurable fields.

We have identified a minimum set of rules that must be followed in the development of picoObjects in any target platform:

- Always be compliant with the standard message format for the communication protocol.
- Only offer support to the simplest protocol version whenever interoperability is not compromised.
- Do not offer support for common middleware services (e.g. Naming and Event services), delegate such responsibilities to other nodes in the network.
- Always use a fully static implementation.
- Resident objects are always on. There is no way to activate or deactivate objects.

The simplest way to achieve a coherent behaviour for each picoObject is by means of a message matching automata. In this context, the allowed set of messages that a certain object understand constitute a BNF grammar defined by the following elements: (1) the message format for the middleware communication protocol; (2) the identity of the object (a unique identifier which is unique in the network); (3) the interfaces exported by the object. It includes the signatures of the provided methods (name, arguments and return); (4) the interfaces that must be inherited from the communication engine and implemented (i.e. Ice::Object is case of ICE); (5) the data serialization rules; and (6) the constraints for the target platform.

Using this information, we are able to automatically generate a fully functional parser whose mission is to identify a whole request message. The corresponding user procedure is automatically invoked and a reply message is generated when a matching happens. If the parser fails to identify a valid method request, then the message is discarded.

Input and output messages are handled on-the-fly using a generated byte-stream processor which saves memory since the incoming message has not even need cached by the device. The request message is processed as the bytes arrive and the reply message is also generated partially from replication of the incoming data. The last part of the reply message (the return value) is generated by the user procedure for each method.

As a proof of the feasibility of our approach, we have developed several prototypes of the picoObjects for two existing middlewares: the already mentioned ZeroC ICE and CORBA. The collection of target platforms and languages used is varied and ranges from assembler for Microchip PICs, Java on a standard embedded PC, Java on an embedded Dallas Semiconductors TINI device, C on a standard embedded PC, C for a Zigbee CC2420 platform, etc.

Table 2 shows an extensive comparative between our picoObjects with equivalent implementations in several commercial platforms. It is important to highlight that our picoObject approach results into solutions that are two orders of magnitude (considering the OS and libraries even more) smaller than their counterparts using commercial solutions.

Middleware	Client	Server (node)	Other	Platform
nORB	-	509000	Libs + OS	PC / Linux
UIC	29000	35000	Libs + OS	WinCE / SH3
UIC	16000	-	Libs + OS	Palm
LegORB	6000	-	Libs + OS	Palm
MQC	14590	22110	JVM + OS	TINI
UORB	45000	45000	JVM + OS	Unknown
Maté	-	16044	OS	MICA / TinyOS
TinyDB	-	58000	OS	MICA / TinyOS
TinyLime	-	16000	OS	MICA / TinyOS
SensorWare	-	237000	OS	IPAQ / Linux
Impala	-	≈ 18000	-	Zebranet
WSP	-	27278	-	Unknown
		picoObjects		
ICE	2 914	3092	-	PIC16F690 / asm
CORBA	2 640	2962	-	PIC16F690 / asm
ICE	-	≈ 6000	OS	PC / Linux / C
CORBA	-	≈ 5000	OS	PC /Linux / Java
CORBA	-	≈ 4096	JVM + OS	TINI /Java
CORBA	-	≈ 5500	OS	PC / Linux / C

Table 2. Minimum server and client sizes on embedded middlewares (the values are expressed in bytes).

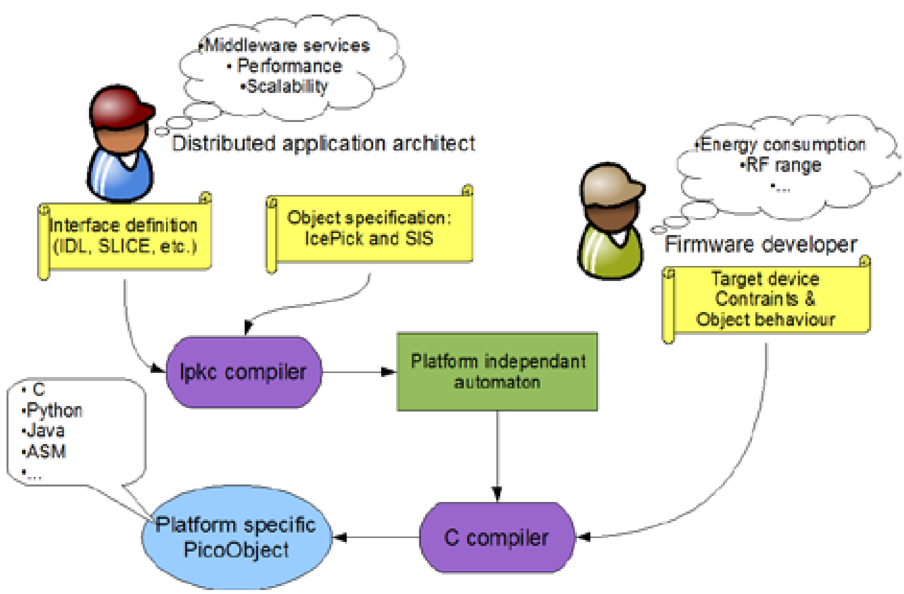

Fig. 1. PicoObject development flow.

The development of picoObjects is supported on OOPAmI by a set of tools and languages that can be used as facilitators by the application developers and embedded code programmers. Figure 1 depicts the whole development flow of picoObjects and the dependencies between the different actors in this process.
First, the distributed application architect must specify the objects present in the system and their relations. Such specification does not only refer to the objects that are going to be implemented as picoObjects in the device but also to those objects that have any relation with them. An interface description language such as IDL or SLICE is used to this end. Along with the operations supported by the objects, it is necessary to provide the object identifiers and the type of relation existent between the picoObject and other objects. For example, a relation cause-effect (i.e. when a picoObject receive a message of type T from X, then reply with a message of type M) or temporal relations (i.e. timers, send periodically a message of type T which is useful for announcement purposes). The IcePick language has been specifically defined for this purpose.
Then, the *lpkc* compiler collects all this information and produces a platform independent definition of the automata (FSM bytecode) that is able to interpret the middleware messages relevant to a picoObject. The automaton, the device restrictions and the implementation of the objet behaviour (provided by the firmware developer) are the inputs for a compiler written in C which produces the final implementation of the picoObject.

3.2 Custom hardware nodes in AmI platform

As we introduced in the beginning of this section, we have versioned an implementation of the system-level middleware entirely in VHDL (Barba et al., 2007). The very first consequence of this fact is that the integration of pure hardware actors[7] in AmI environments is performed in a seamless way.
In the various demonstrators we have built to prove the transparent communication between our hardware nodes and external components, we take as the reference the ZeroC ICE object-oriented commercial middleware widely used in the industry.
The base architecture of a hardware node in OOPAmI is shown in Figure X and comprises:

- The *objects*, implemented as hardware units. A single node can group several of this HwOs, enabling resource sharing and thus, reducing the cost of the final implementation. HwOs can be implemented as static objects or dynamic objects using reconfigurable logic. Dynamic objects can be instantiate and evicted into the reconfigurable area at run time. All the tasks concerning the management of the reconfiguration process are performed by the *Reconfiguration* of the middleware.
- The *External Object Adapter (EOA)*. It provides connectivity with external objects and/or systems. One side of the EOA depends on the component that acts as the bridge with the external network (Remote Network Interface) so it has to be hand made. However, on the side interacting with the local on-chip network (if it is present), the control logic is fully customizable and generated in an automatic way.
- The *interconnection fabric*. When there is more than one HwO embedded in the node, the in-chip communication infrastructure allows the connection of the HwOs

[7] The presence of a processor, running the operating system or a software controller routine is optional.

with the EOA (if not, such connection is done through a point to point link). Also, the HwOs may use the bus, Network-on-Chip or any communication architecture present inside the chip to interact.

In other approaches, since a common communication infrastructure is missing, on-chip functionality may only be accessed from off-chip components using an ad-hoc interface that exists only if it has been predicted by the designer.

The External Object Adapter

The EOA has mainly two duties, namely: 1) translates the ICE object identifiers and operations (strings) into internal local addresses; and 2) adapts the ICEP (ICE protocol) message format to the in-chip message format. Figure 2 represents the place that the EOA occupies in the hardware node architecture. Next, we detail the role of each functional unit using the two possible communication scenarios:

- Incoming message treatment. The *packet analyzer* examines the incoming frames and throws away those that are not valid ICEP messages. The UDP/TCP listening ports and the allowed sources (Internet Protocol addresses) can be configured in the EOA. The *packet sequencer* module internally caches some parts of the ICEP message using the *in-progress request info memory* in order to build a later response message. Finally, the packet sequencer injects the message body "as is" without pre-processing the data because the total compatibility of the coding rules (as we detail below). The target HwO is addresses using the information maintained by the *external routing table* (ERT), the packet analyzer uses it to translate the ICE identifiers.

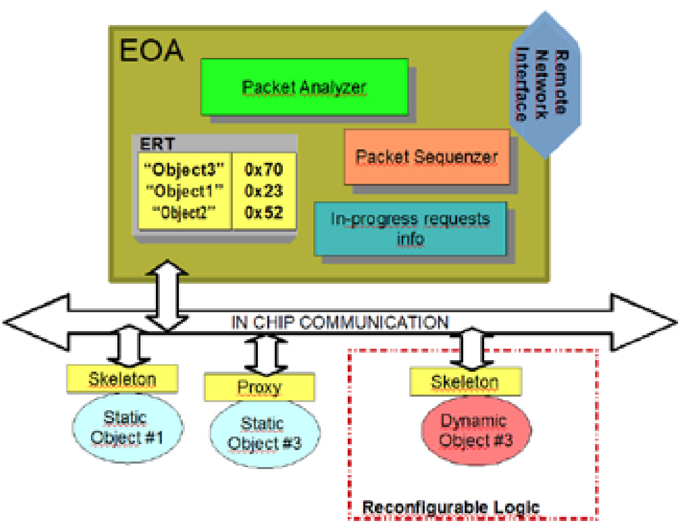

Fig. 2. OOPAmI hardware node architecture.

- Outgoing message treatment. When a in-chip transaction is addressed to an object that is not implemented in the hardware node (i.e. "Object2" in figure 2) the EOA routes the traffic to the external network. The *sequencer* fills a frame header template (it depends on the external network packet format) with the valid ICE identifiers, external network address, etc. The needed information can be retrieve from the ERT (for a request) or the data previously cached (for a response). Finally, the frame is sent throw the remote network interface.

Hardware Objects (HwOs)

A HwO is the logic that implements the functionality of the equivalent software object plus the logic of the wrappers (the hardware version of classic *proxies* and *skeletons*) that are in charge of translating in-chip transactions in *local invocations* to the HwO. We intentionally decoupled communication from behaviour (following the *Remote Method Invocation* semantics) to make HwO modules reusable in future designs. By isolating HwOs from communication implementation details we make them immune to unforeseen changes in the communication infrastructure.

Both proxies and skeletons agree in how method invocations translate in a sequence of low level actions (write and read primitives[8]) over the interconnection architecture to activate the execution of the operation. Actually, at the logical level a method call results into an exchange of message between the initiator and the target (HwOs can be either initiators or targets of a method invocation).

The RMI protocol for hardware implementations defines the format of the message to be sent and how to code the arguments of both invocations and results. For example, the header fields of a message are combined to form the address of the destination of a transaction.

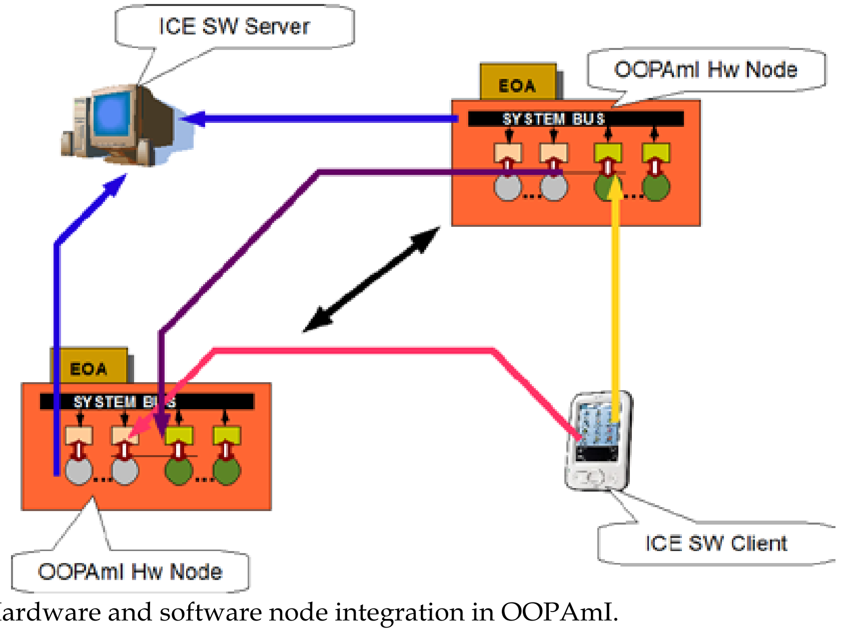

Fig. 3. Hardware and software node integration in OOPAmI.

[8] They are basic services offered by most of the buses so that we do not limit the platform implementation to a concrete technology.

But what really provides the inter-component communication semantics with standard ICE software clients or server as well as with other hardware nodes is the use of the same data type system and coding rules. HwOs support all basic types (bool, short, int, float, etc.) plus structures, sequences (vectors) of a fixed size and any combination of them. This makes our hardware implementation of the middleware 100% compatible with a well known subset of ICE features. The encoding rules defined by ICE are quite simple in order to propagate such simplicity to the architecture elements that will manage the marshalling and unmarshalling processes (proxies and skeletons). This simplicity allows reaching an efficient component design with the minimum overhead.
Thus, the format of the body messages remains unchanged, no matter the nature of the communicating objects, which means that a target HwO is not able to distinguish whether the source of the invocation is a SW or another HwO.
In Figure 3, four interaction scenarios are depicted involving hardware software nodes. It is shown how existing on-chip objects can communicate with SW servers (blue line) and server objects implemented (black and purple lines) as HwOs. Also, HwOs can be accessible from outside (yellow and red lines).

Development flow
Along with the definition of an ICE compatible hardware platform architecture, we have developed a design flow to easily create and integrate HwOs. Figure 4 shows the workflow of the proposed HwO generation process. The input to this process is an object interface description, written in an interface description language (SLICE in the case of ICE), just like the other objects will see the hardware component in the system. This information is used to generate: (1) the on-chip communication infrastructure and (2) the final hardware object implementation.

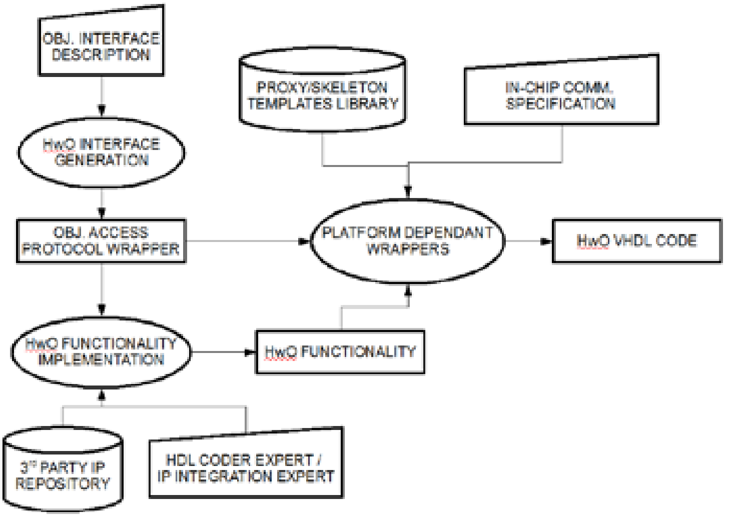

Fig. 4. Design flow of HwOs in OOPAmI

This HwO implementation can be the result of an integration process, reusing existing designs implementing the required functionality. A new, hand coded chip implementation is needed when no reuse opportunities are present for the current application.
Since we have standardized the way the functionality of a HwO is invoked (HAP, the *HwO Access Protocol*[9]), we are able to automatize the generation of the proxies and the skeletons (a critical step in the hardware design flow). The final implementation is highly optimized to fit the low-cost requirements while keeping the process automatic to save design time.

4. Middleware Services and advanced features

In this section, we briefly describe those services and advanced features that differentiate OOPAmI from other middlewares. The AmI service developer point of view drives the middleware design process. We use the desirable characteristics that a global service development process should have as the guidelines for middleware design (with independence of software engineering methodology applied).

4.1 Information model

Firstly, the developer needs to know what services from the middleware can use in order to retrieve the information is going to be consumed by a new service. In the same way, only in the case the new service generates information, the developer has to know how this information must be generated (e.g. the format of the data and how the data is exported to the rest of services). Also the developer should know the interface to access other services and their attributes (including, valid values and format for these attributes).
Our information model includes the items listed above It is basically a taxonomy of services (specialization of generic services) that includes attributes for each service. This taxonomy is like our yellow pages of services and permits developers to lookup for service classes and its attributes. Each service instance running in an AmI environment belongs to a class expressed in the taxonomy and instantiates the attributes of this class. We consider this taxonomy as a mean to establish a common nomenclature for AmI service developers (aiming the same goal POSIX interfaces play in operating systems).
If we want to develop a new service (for example, modelling a new type of camera, light, virtual glasses, etc.), we should look up in the taxonomy, find the class that better fits such new service and follow the nomenclature throughout the rest of the service development process.
Ontologies are used to express the taxonomy because it is a powerful tool to visually organize information and the existent relations between the entities in the information model. Along with our ontology we define a core of basic interfaces that are used to model any service in the ontology. By aggregation, we are able to generate more complex interfaces that offer read and write operations for each attribute. The Interface Definition Language (IDL) allows us to express interfaces in a clear language (from the developers point of view) avoiding the complexity of other languages (for example WSDL for web services).
Some services need more advanced interfaces than read/write primitives. In many cases, they follow standards adopted by the industry as in the case of AVStreams from *Object*

[9] This also includes the physical interface of the hardware module (Jesús B. et al. IPSOC 2006).

Management Group (OMG, 2000a) for multimedia stream configuration, Mobile Location Protocol (Open Mobile Alliance, 2001) for spatial coordinate information and property services (OMG,2000b) for attribute management (used when low cost devices has to delegate its management to more powerful devices).
A set of basic events is also specified in the information model. With the classification of the events that can be generated in an AmI environment, the service developers have a common reference that can be use to orchestrate common behaviours. Of course, not all services will consume or generate all events. The purpose of this definition is to let the designer know a list of common events that a service could receive or generate.
The next step is to know how a service can be accessed from other services. Since we use an object model, each service has a reference which is actually a pointer to the object that implements such service. This mechanism does not differ from traditional object oriented languages but the need to get the service reference before use it.

4.2 Abstract Service Discovery Protocol (ASDF)

To allow developers to find any reference to any service in the AmI environment (note that the services are deployed in a highly distributed environment) we have defined an interface for service announcement and discovery. This ASDF interface is the result of thorough study of the service discovery protocols (SDP) most widely used in the industry and looks as follows:

```
module ASD {
        dictionary<string , Object> PropDict;
        interface iListener {
                void adv(Object_ prx , iProperties : : R_ prop) ;
                void bye(Ice : : Identity oid) ;
        } ;
        interface iSearch {
                void lookup(Object_ cb , PropDict query) ;
        } ;
} ;
```

As the reader can see the interface is clear and simple to understand (the same interface in WSDL would take almost one page in XML with multiple references).
There are two interfaces that conforms the ASDF definition: *iListener* for announcements and *iSearch* to lookup new devices supporting specific services. It is not mandatory for a device to implement the two interfaces (for example a basic sensor probably only sends announces to the environment and it does not need to lookup for nothing).
Those services which send an advertisement about its presence (invoking the *adv* operation) attach to this invocation their references (*prx*) and a reference to the service that manages its properties. In the case of services conceived to run in powerful devices, the interface *iProperties* can be instantiated. The properties interface defines the capability of a device to manage its own attributes. If this interface is not present (i.e. in the case of less powerful devices) such responsibility is delegated to other object in the system so a reference to it must be provided (the process is transparent to the clients). The later scenario is very common since it is highly flexible.
When a service invokes an adv operation, it propagates an event to a well known event channel (labelled as "ASDA") where other services in the AmI environment have previously

subscribed. The management of this event channel is provided by the middleware and can be configured with replication (for fault tolerance), with QoS parameters and federated associations for scalability purposes.
In the case of a lookup operation, the service interested in finding other services creates a temporal event channel. Then, the services that fit the properties included in the *query* parameter send an advertisement. We offer the option of a repository of services that receives all adv announcements and answers to lookup operations.
With ASDF, the developer does not need to configure any fixed reference to other services and only needs to lookup for the desired services. Sending advertisements to well event channels is the mechanism implemented to offer the functionality of a service to the rest of environment.

4.3 AmI simulator

Testing non trivial AmI services can be difficult due to the heterogeneity of possible AmI environments and situations that can take place in this environment. One of our actual ongoing works is the creation of an integrated simulation environment including:

- A set of dummy services that implement real interfaces.
- An AmI Specification Language (AmISL) to model:
  - The physical environment with the device placement.
  - Human entities and their behaviour inside the virtual environment (movements, actions and interactions activities).
  - The state of the environment (temperature, light, state of doors, etc.)
- A simulation engine that interprets the actions expressed in AmISL. For example, the synthetic data generation simulating sensor activity.
- A log tool that records the information of the service that is being tested, focusing on its interaction with the dummies services.

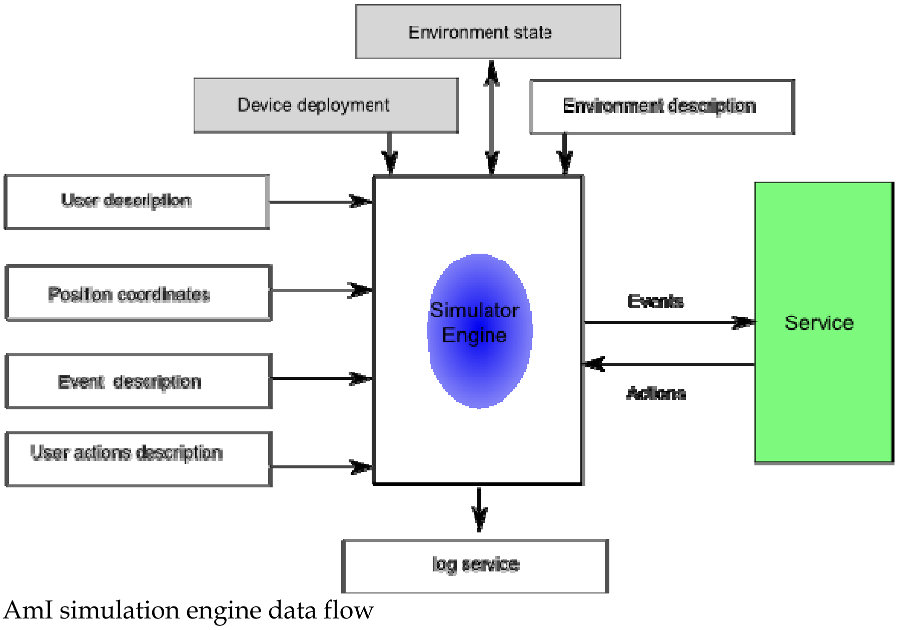

Fig. 5. AmI simulation engine data flow

In figure 5 we can see the information flow involved in the simulation process.
Once the service has been simulated and tested, the developer needs to deal with problems like *how do I deploy/stop/play/actualize my service* in the environment or *what do I have to do to obtain the initial references to the basic middleware services* (i.e. event channel service).

4.4 BootStrap service

The *bootstrap* service has been developed for easy integration of services and devices. When a device is begin deployed in any environment, it is necessary to do some tasks before offering the functionality to the rest of services in the AmI environment. With independence of the procedure used by the device to get network connectivity, the bootstrap service starts automatically a broadcast query looking for an entity called *environment manager*, this environment manager maintains the references to basic services. Each device runs a *service manager*, a directory of the services that run in this device.
The environment manager runs in a node labelled as the *coordinator*. The bootstrap service is a distributed algorithm for choosing the *coordinator* and its *replicas*. Each candidate replica can promote to the *coordinator* role when the coordinator fails.
When the *bootstrap* service of a new device starts to run, the *bootstrap* service tries to identify the coordinator of the environment and gets the reference to the environment manager. With this reference the *service manager* is then set up, the references to the basic middleware services are retrieved and the local services start to run.
As in the ASDF service, the *bootstrap* service tries to minimize configurations procedures for service and device integration using a Place&Play philosophy, similar to the successfully Plug&Play followed by the hardware industry.

4.5 Transparent dynamic reconfiguration

The reconfiguration service in our middleware has the capability of instantiating new hardware services in a transparent manner (Rincón et al., 2009).
A reconfiguration procedure implies the modification of the physical interface of the module to be reconfigured, so we need isolate the reconfigured area. Following the distributed object oriented paradigm, the hardware version of proxies and skeletons enable us to have a fixed interface (the one with the bus or network interface) so we can instantiate into any reconfigurable area.
Other problem related with reconfiguration is state persistence. Due that we know the variables that have all information related with the state of the object and its size (at design time) we know the information that we have to save and restore.
We extend the hardware adapters that translates bus read and write operations into object invocations with the following operations: *stop*, *start*, *getState*, *setState* and *initState* which are used for start, stop, get and set the state and reset a hardware object respectively.
Inside of a hardware node with reconfiguration capability (for example, an FPGA) we define a set of entities and services for a transparent reconfiguration procedure:

- *Memory Allocation Service*: this service is a centralized memory management entity for the whole system that presents a well known interface, completely independent from a concrete implementation technology or memory hierarchy.
- *Object Location Service*: the location service contains a table of references where hardware object identities are linked with valid endpoints.

- *Object Factory*: the factory service physically instantiates an object into a reconfigurable area. We can create objects of any type at run time providing its class type and a reference to the memory location of the partial bitstream (the binary image of the code).
- *Reconfiguration controller*: this entity has the responsibility of run-time creation and destruction, including object state persistence management. This service is built upon the three basic ones described before (memory allocation, location and factory).

Each of these services are objects themselves. This means that they can be implemented as HwOs or software entities. Even due to the external communication capabilities of a hardware node the reconfiguration process can be initiated and monitored from outside the chip. A possible organization will leave the reconfiguration controller and location service outside the hardware node whereas the factory and memory allocator services are attached to the hardware implementation for the best performance.

The reconfiguration process can take place explicitly or implicitly. The former is the case of an implementation of a migration service (or a application) that schedules the instantiation of HwOs and their movement across different hardware nodes. The later happens when a method of a dynamic HwO is invoked and it is not loaded in any reconfigurable area. In this case, the reconfiguration controller takes the control and instantiate the HwO without the intervention of the application that generated the invocation.

4.6 Location service

There are several position systems based in different technologies (Wifi and Bluetooth cells, RFID technology, etc.) for indoor scenarios. These systems can be combined to potentially improve the accuracy of positions of people present in the environment. Our location service combines all these systems (and the outdoor facto standard, the Global Position System) with the *Mobile Location Protocol* (Open Mobile Alliance, 2001) in order to provide a service for people location and identification. We define a set of interfaces to add new positioning systems and a set of rules to combine the information coming from such systems.

For example, the Bluetooth interface present in mobile phones can give us a clue about the area where the owner of the mobile phone is present. We can also combine the Bluetooth information with movement sensors, spread in the environment, in order to improve the accuracy of our location service. The defined interfaces enable us to use events from the environment as triggers to notify variations in the scene. For example when a door is opened or closed, a desktop session in a computer begins, the turning on or turning off of devices, detection of faces, etc.

In the location service design, we emphasize the modularity of the system in such a way that we use plug-ins to provided new functionality and “inference” rules about location coordinates.

4.7 Service Composition

Providing service composition in an autonomous manner is not trivial, and it is not achieved yet without involving users, at some level. The approach integrated in OOPAmI proposes a multidisciplinary approach, in the shape of a layered architecture. Founded on the aforementioned middleware platform, a multi-agent system is deployed on top of it. These intelligent agents retrieve information from the context, in order to provide the reasoning engine with the context information required to provide an adequate response to the current situation. This response will be provided in terms of the basic services, that when combined in a plan, will outcome the composite service.

The combination of these different technologies is successfully achieved by counting on a unique semantic model, implemented by the middleware framework, the multi-agent system and the reasoning engine. This semantic model, or ontology for a service-oriented architecture, basically considers the following entities: device, service, action, object, and property. *Devices* provide *services*, and are described in terms of *properties*, such as their location, or provided features. *Services* are described in terms of *actions* performed over *objects*, and also hold *properties.*

An OWL description of this semantic model is translated into ICE interfaces, so that all services provide a common access method. The same OWL description is used by the multi-agent system, not only for message exchange, but for interacting with the middleware services. Finally, the reasoning engine resorts to the same semantic model to describe the domain knowledge for the deployed context. Therefore, the results of the inference and search processes can be carried out by the multi-agent system.

5. Conclusions

In this chapter we have shown our middleware guidelines for a holistic approach to AmI environment development. The object oriented paradigm drives the development of the middleware services making it easier their utilization and modelling. Besides, the integration of heterogeneous devices is performed in a seamless way.

The completeness of our approach includes service development support, service discovery facilities, a tool-chain for the most complicate tasks in the case of HwOs and embedded software generation, a simulation framework and more.

Current work is focused on the extension of the pool of services that can be offered to the industry and service developer’s community in order to increase the capabilities of the middleware.

6. References

Barba J.; Rincón, F.; Moya, F.; Dondo J.; Villanueva, F.J.; Villa D. & López J.C, (2007). OOCE: Object-Oriented Communication Engine for SoC Design. *Proccedings of 10th EUROMICRO Conference on digital System Design*, pages 296-302, ISBN: 978-0-7695-2978-3, August 2008.

Friedewald, M. & Da Costa, O., (2003). Science and Technology Roadmapping: Ambient Intelligence in Everyday Life, technical report, JRC/IPTS European Science and Technology Observatory. Available online at:: http://www.cybertherapy.info/pages/AmIReportFinal.pdf (last visit Sept. 15th 2009)

Gelissen, J. (205). IST-Ozone Project Final Report, technical report, Available at : http://www.hitech-projects.com/euprojects/ozone/public_docs/ozone-phr-19Jan05-final-report-jg.pdf (last visit Sept. . 15th 2009).

Henning, M. & Spruiell, M., (2008). Distributed Programming with ICE, version 3.3. At http://www.zeroc.com (last visit Sept. 15th 2009).

Issarny, V. ; Sacchetti, D. ; Tartanoglu, F. ; Sailhan, F. ; Chibout, R. ; Levy, N. & A. Talamona (2005). Developing Ambient Intelligence Systems: A Solution based on Web Services. *Journal of Automated Software Engineering*, Vol. 12, Issue 1, pages 101-137, January 2005, ISSN : 0928-8910.

IST Advisory Group (2003). *Ambient Intelligence: from vision to reality*, technical report. At: http://www.cordis.lu/ist/istag-reports.htm (last visit Sept. 15th 2009).

Maly, I. ; Curin, J. ; Kleindienst, J. & P. Slavik. Creation and Visualization of User Behavior in Ambient Intelligent Environment. *Proccedings of 12th International Conference Information Visualization*, pages 497-502, ISBN: 978-0-7695-3268-4, July 2008, IEEE Computer Socity, Whashington DC.

Marsa, I. ; López-Carmona M.A & Velasco, J. R. (2007). A hierarchical, agent based service oriented architecture for smart environments *in Service Oriented Computing and Applications*, Springer, ISSN : 1863-2386, London.

Object Management Group (2000). *Audio/Video Stream Service*. OMG Document formal/00-01-03.pdf, 2000a.

Object Management Group (2000). *Property Service Specification*. OMG Document formal/00-06-22.pdf, 2000b.

Open Mobile Alliance (2001). *Mobile Location Protocol Specification*. Document Location Interoperability Forum TS 101 Specification, version 3.0.0. Available online at: http://www.openmobilealliance.org/tech/affiliates/lif/lifindex.html (last visit Sept. 15th 2009).

OSGi Alliance (2006). *OSGi Service Platform: Core Specification*, edition 4.0.1 Release 4, July 2006. At : http://www.osgi.org/Release4/Download (last visit Sept. 15th 2009).

Perumal, T. ; Ramli, A.R. & Yew Leong, C. (2008). Design and Implementation of SOAP-Based Residential Management for Smart Home Systems. *IEEE Transactions on Consumer Electronics*, Vol. 54, Issue 2, pages 453-459, ISSN : 0098-3063.

Ramos, C ; Augusto, J.C. & D. Shapiro (2008). Ambient Intelligence- the next step for Artificial Intelligence. *IEEE Intelligent Systems*, Vol. 23, Issue 2, pages 15-18, March-April 2008, ISSN : 1541-1672.

Rincón, F.; Barba J.; Moya, F.; López J.C & Dondo J. (2009). Transparent Dynamic Reconfiguration as a Service of a System-Level Middleware, *in Lecture Notes in Computer Science*, Springer, ISSN: 0302-9743, Berlin.

Sacchetti, D. ; Bromberg, Y.-D. ; Georgantas, N. ; Issarny, V. ; Parra, J. & R. Poortinga (2005). The Amigo Interoperable Middleware for the Networked Home Environment. At : http://middleware05.objectweb.org/WSProceedings/demos/d4_Parra.pdf (last visit Sept. 15th 2009).

Villanueva, F.J.; Moya, F.; Santofimia M.J.; Rincón, F.; Villa D.; Barba J. & López J.C, (2009). Towards a Unified Middleware for Ubiquitous nad Pervasive Computing, I*nernational Journal of Ambient Computing and Intelligence*, Vol. 1, Issue 1, pages 53-63, ISSN: 1941-8647.

Wu C. & Fu, L. (2008). A Human-System Interaction Framework and Algorithm for UbiComp-Based Smart Home. *Proceeding of 2008 Conference on Human System Interaction*, pages 257-262, ISBN: 978-1-4244-1542-7, May 2008.

3

A review of indoor localization technologies: towards navigational assistance for topographical disorientation

Jorge Torres-Solis, Tiago H. Falk and Tom Chau
Bloorview Research Institute & University of Toronto
Canada

1. Abstract

Indoor localization technologies hold promise for many ambient intelligence applications, including in-situ navigational assistance for individuals with wayfinding difficulties. Given that the literature on indoor localization is vast and spans many different disciplines, we conducted a comprehensive review of the dominant technologies. We propose a taxonomy of localization technologies on the basis of the measured physical quantity. In particular, we identified, radio frequency, photonic, sonic and inertial localization technologies as leading solutions in the field. For each selected technology, the fundamental scientific mechanisms for localization are explained, key recent literature appraised and the merits and limitations are discussed. Recommendations are made regarding the creation of context-aware systems that can be used to enhance a user's topographical orientation skills.

2. Acronyms

AoA angle of arrival
BLUPS Bluetooth and Ultrasound Positioning System
DS/CDM Direct Sequence Code Division Multiplexing
DToA difference of time of arrival
GPS Global Positioning System
HMD Head Mounted Display
IR Infrared
IrDA Infrared Data Association
LOS Line-of-sight
NLOS Non-Line-of-sight
PDA Personal Digital Assistant
RDS root-mean-square delay spread
RF Radio Frequency
RFID Radio-frequency Identification
RSS Received Signal Strength

ToA time of arrival
ToF time of flight
TD Topographical Disorientation
TV television
UWB Ultra-Wideband
WLAN Wireless Local Area Network

3. Introduction

Topographical Disorientation (TD) refers to a family of deficits in environmental orientation and navigation. Aguirre and D'Esposito [3] provide a well-accepted taxonomy of TD, arguing that difficulties in wayfinding may arise as a result of the combination of different cognitive impairments. For example, it is well recognized that TD and spatial navigation deficits are common sequelae of brain injury [87, 74]. Individuals living with post-traumatic effects of brain injury are oftentimes faced with symptoms such as weak visual scanning skills, or deficits in complex attention, prospective memory or sequential processing [47]. These symptoms cause problems of interaction with and perception of the surrounding environment even several years post-injury [16, 72]. It has also been argued that deficits in topographical orientation can lead to spatial anxiety or wandering behaviours [11, 77, 87].

It has been suggested that wearable navigation technologies such as Global Positioning System (GPS) can be a useful wayfinding tool for individuals with cognitive impairments [7]. However, GPS signals have limited coverage indoors (e.g., [36, 40, 30, 61, 99, 38]). Given that patients spend significant periods of time indoors - be it in acute and tertiary care hospitals or, subsequent to rehabilitation, at home, schools, office buildings, shopping malls, long-term care facilities - identification of potential technologies for indoor navigational assistance is imperative. An initial survey of the literature has suggested that a diverse collection of candidate indoor localization technologies exists across many different disciplines. This diversity makes it difficult to grasp the potential of an existing technology for the rehabilitation of individuals with topographical disorientation.

Localization technologies are critical to emerging location-aware guidance systems and support services for individuals who have wayfinding difficulties due for example to low vision [69], stroke [86] and traumatic brain injury [6]. In particular, regarding indoor navigation systems for individuals with topographical disorientation, localization has often been human-mediated rather than automatic. For instance, Liu et al. evaluated the benefits of navigational tools in real indoor environments [46]. However, the location tracking and tool display decisions in their experiments were not automatic, but controlled by the experimenters. In similar vein, Sohlberg et al. [75] found that individuals with wayfinding difficulties secondary to brain injury responded well to speech-based auditory directions from a wrist-worn PDA navigation system. However, like Liu et al., the PDA's navigational instructions were transmitted by a human operator at a mobile computer. Undoubtedly, there is immense opportunity to explore the potential of automatic patient indoor localization technologies in the emerging fields of cognitive prosthetics and situated assistive technologies. As a consequence, the overarching goal of this review is to systematically organize the literature on indoor human tracking technologies, and to ascertain their feasibility for eventual use in the realm of TD rehabilitation.

4. Literature selection

We combed the literature for candidate localization technologies that could serve to create an assistive device for individuals with TD in indoor environments. In particular, peer-reviewed journal articles published in English between 2003 and 2009, inclusive, were sought from three different academic databases, namely: Compendex, Inspec and Geobase, using the keywords, "*indoor location*", "*indoor localization*", "*indoor tracking*" and "*indoor positioning*". After removing duplicate records, we arrived at 214 articles. To identify potential technologies applicable to the creation of a navigational assistance device that for individuals with topographical disorientation that offered accurate information in real time, the returned articles were subsequently screened according to the following inclusion criteria:

1. The article must focus on the development and experimental testing of a localization or navigation system: i.e., articles focusing on mathematical processing of localization data, or localization experiments in simulated environments were discarded.
2. The reported technology must:
   (a) be usable indoors, within a building or a larger space, i.e., technologies used to track a capsule inside the human body, or a device within a single room were excluded;
   (b) offer a localization accuracy of a mobile target within a 10 meter radius with a delay of 5 seconds or less;
   (c) be applicable to humans, i.e., systems designed for vehicles, large objects, or objects that relied on a fixed pose or odometry measurements of a robot were excluded; and
   (d) track and identify multiple humans concurrently.

Fifty three articles met such initial criteria. Such articles were subsequently scanned for alternate localization technologies that were referenced three or more times and that were not selected in the initial search. Eleven additional articles were included in this manner, totaling sixty four articles for consideration in the present review.

4.1 Taxonomy of localization technologies

The location of an object in space is determined by measuring a physical quantity that changes proportionally with the position of the object of interest. The present review is structured in terms of the measured physical quantity. The selected articles were divided into six main categories based on the physical quantity measured, namely, (1) radio frequency waves, (2) photonic energy, (3) sonic waves, (4) mechanical energy (inertial or contact), (5) magnetic fields, and (6) atmospheric pressure. Each physical quantity grouping can be further subdivided according to the underlying hardware technology. Figure 1 summarizes this two-tiered taxonomy. Note that the latter two phenomena (magnetic and atmospheric) have been collapsed into one category, named, "Other" due to low article counts in these areas.

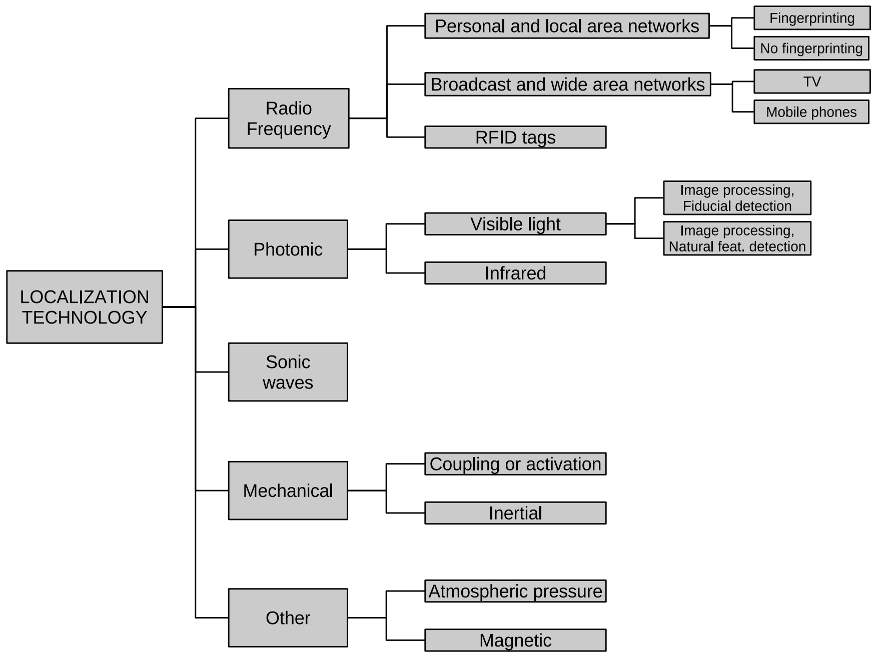

Fig. 1. Taxonomy of indoor localization technologies by measured physical quantity and hardware technology

Where appropriate, articles are further differentiated by principal localization technique. For this third level of classification, the following localization technique definitions are provided, expanding on those proposed by Hightower and Borriello [29]:

Triangulation is a family of methods that include lateration, angulation and variations thereof. Lateration refers to the calculation of the position of the human subject based on his relative distance to several previously-known fixed points in space. Such distances are commonly obtained indirectly by measuring parameters that are proportional to distance. Time of flight and power attenuation of a radio signal are common indirect distance metrics [20]. Angulation refers to the calculation of the position of the subject using the angles of arrival of signals emitted from fixed points in space. [88, 80].

Proximity refers to a class of methods which establish the presence of the human subject in the vicinity of a sensor, which alone has limited sensing range and analysis capabilities. The proximity of the subject can be detected through physical contact, presentation of a device such as a magnetic band to an appropriate reader or through the monitoring of a physical quantity in the vicinity of the sensor, for instance, a magnetic field.

Scene analysis involves the monitoring of a wide area around the subject of interest from a specific vantage point. The commonly deployed sensors have broad coverage area and range. Examples include ceiling-mounted video cameras or passive infrared (PIR) sensors.

Dead reckoning refers to the usage of sensors that provide location updates, calculated using information about a previously-estimated location. Position estimation is commonly based on accelerometry and gyroscopy.

The ensuing review of literature will adhere closely to the taxonomy depicted in Figure 1. For each physical phenomenon, we will briefly present the general principles of localization, review articles in the relevant subcategories and comment on their relative merits and limitations. We will conclude the article with recommendations of indoor localization technologies suitable for addressing the development of assistive devices for individuals with TD.

5. Radio Frequency

An electromagnetic wave is the energy generated by an an oscillating, electrically charged particle in space. The generation of electromagnetic waves is known as a radio frequency emission.
Solutions in this category estimate the location of a mobile target in the environment by measuring one or more properties of an electromagnetic wave radiated by a transmitter and received by a mobile station. These properties typically depend on the distance traveled by the signal and the characteristics of the surrounding environment.
As depicted in Figure 2, most of the articles in this survey describe a Radio Frequency (RF) localization system. These articles can be further subcategorized according to the underlying hardware technology as listed below.

1. Personal and local area networks, including technologies such as IEEE 802.11, Ultra-Wideband (UWB), ZigBee, or Bluetooth, either as the sole localization technology [12, 92, 17, 43, 20, 64, 66, 42, 41, 53, 79, 28, 22, 96, 18, 67, 4, 95, 88, 23, 83, 9, 57] or as a contributing technology within a hybrid solution [98, 13, 100, 49, 60, 2].

2. Broadcast and wide area networks, including networks designed for localization purposes, such as the GPS and the Global Navigation Satellite System (GNSS) [67], and broadcast networks not originally intended for localization purposes, such as cellular phone networks [73, 25] and television broadcast signals [62, 63].

3. Radio-frequency Identification (RFID) tags [45, 35, 81]

4. Radar [65, 71]

Each flavour of RF localization is reviewed below.

5.1 Personal and local area network-based solutions

5.1.1 Fingerprinting-based localization solutions

Most articles included in this section propose the localization of mobile targets using a two stage process [98, 17, 64, 66, 42, 53, 9, 57, 92, 43, 79, 22, 96, 18, 67, 4, 95, 83]. The first stage consists of an off-line radio scene analysis (i.e., performed prior to localization) in which a mobile station extracts radio fingerprints, i.e., features from one or more metrics of the radio signal measured at predefined points in the environment. These radio fingerprints are proportional to the distance between the mobile receiver and the emitting station. Common

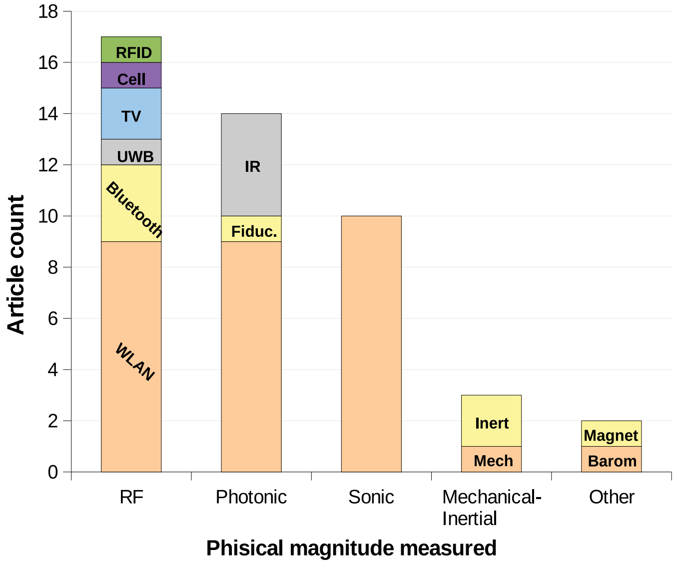

Fig. 2. Distribution of articles by physical quantity measured

metrics include the direction or angle of arrival (AoA), Received Signal Strength (RSS), or time of flight (ToF) of the incoming radio signal [52]. A radio map or database of fingerprints is created, storing signal feature values at each location along with the corresponding spatial coordinates. Some authors propose automated or assisted radio map creation techniques, exploiting the characteristics of the environment, such as the spatial configuration and the material composition of the environment [79, 34, 98].

The on-line stage comprises the active localization process where the mobile receiver extracts a fingerprint of the radio signal at an unknown location. Localization is commonly achieved by proximity techniques, i.e., finding the closest match between the features of the received radio signal and those stored in the radio map [98, 17, 64, 66, 42, 22, 18, 67, 4, 57, 92]. More accurate localization can be achieved using a triangulation-like process, in which several candidate locations (each with a fingerprint bearing some resemblance to that of the received signal) are geometrically combined to provide an estimate of the receiver location in space [43, 53, 79, 67, 95, 83, 9]. Algorithms deployed in the selection of the closest match or matches from the radio map include: 1) nearest neighbours techniques and variations thereof [17, 95, 9]; 2) Bayesian statistical matching [98, 92, 43, 64, 66, 67, 83, 57]; 3) maximum likelihood estimation [79, 22]; 4) correlation discriminant kernel selection [42, 53]; and neural networks [18, 4].

Some fingerprinting techniques also provide coarse estimates of orientation, for example, 4 different orientations [92, 9]. The radio map is created with a user transporting the mobile device. At a given location, a fingerprint is recorded at each possible orientation. Since the human body affects the propagation of radio signals, the fingerprint generated for each orientation will be different [92, 9].

Fingerprinting localization accuracy is commonly down to a few meters (i.e. within 3 metres 90% of the time [42] and within 3 metres 91.6% of the time [53]). The lowest number of base stations used to create the feature space of a fingerprint was one [98]. In other words, fingerprinting seems to provide reasonable localization accuracies without excessive hardware requirements. The most pressing challenge however is the non-stationarity of the radio map. This is reflected as differences in the measured signals during the on-line and off-line phases at the same exact location. The time-variant nature of the radio map can be attributed to radio signal propagation effects induced by dynamic aspects of the environment such as the presence or absence of people, elevators, moving doors and other environmental changes [45, 42, 9].

As a particular case, in [67], the authors proposed an indoor and outdoor hybrid localization system, combining GPS and WLAN localization technologies. The authors proposed handing off the localization responsibility between GPS and WLAN depending on their availability. Fingerprinting was done through a GPS-on-line stage, collecting positional information of the access points within a nearby building and geo-referencing these measurements with the information obtained through GPS. After this fingerprinting process, indoor WLAN positioning was achieved by estimating a point located among a set of the most probable locations. These locations were predefined by the user - i.e. copy room, cafeteria. The histogram of the RSS measurements was used to determine the location that best matched the histogram received. If the histogram did not closely match any known locations, a centroid algorithm estimated the location of the user from the locations of previously geo-referenced access points and probable nearby locations. The place detection algorithm, which relied on an almost perfect fingerprint match, yielded room level localization accuracy, while the WLAN localization algorithm, using the centroid of several probable locations, yielded an accuracy of approximately 30 metres.

5.1.2 Non-fingerprinting-based solutions

RF-based localization can also be achieved without *a priori* analysis of the radio properties of the environment (i.e., without development of a radio map). Four of these articles, all of them based on UWB radio signals, rely on signal triangulation as the sole localization technique [28, 88, 23, 41], while in [20] localization is achieved by proximity and scene analysis.

Indoor localization based on triangulation of radio waves is a non-trivial problem because the transmitted signal can suffer obstructions and reflections. As a consequence, Non-Line-of-sight (NLOS) conditions emerge. In the presence of NLOS conditions, the radio signal can travel to the receiver through a non-direct path, giving rise to erroneous distance estimates.

To overcome these problems, the use of UWB radio signals has become the most novel solution in radio frequency-based solutions. The properties of ultra-wide band, short duration pulses mitigate the propagation problems associated with multi-path radio propagation. The most representative example is the system proposed by Venkatesh and Buehrer. They introduced a triangulation localization system based on impulse UWB radio signals [88]. They suggested that the statistical parameters describing the distribution of the received root-mean-square delay spread (RDS) serve as the best discriminant estimator between Line-of-sight (LOS) and NLOS signal propagation. This means that the statistical parameters defining the RDS of the received signal can be compared against a predefined rule set to determine if the signal was received via a direct or indirect path. Subsequently Venkatesh and Buehrer tracked a mobile station though 71 predefined locations within a building, achieving localization accuracies ranging from 1 centimetre to 2 metres. As another example of RF triangulation-based schemes, Krejcar and Cernohorsky presented a localization system [41] that relied on the triangulation of RSS metrics. Room-granularity accuracies were reported but further details of the triangulation or localization schemes were not revealed.

Finally, a system based on a combination of scene analysis and proximity techniques using a Bluetooth ad-hoc network was presented in [20]. Bluetooth inquiry signals were used for localization. In inquiry mode, a bluetooth device inquires about neighbouring bluetooth stations. This inquiry process consists of scanning for devices in the vicinity, using a sequence of different power levels. Low power levels will detect devices in close proximity while high power levels will include devices that are located farther away, providing coarse distance estimates in this fashion. This approach requires a fixed or "anchor" node which establishes the position of nearby mobile nodes. Subsequently, the localized nodes can establish the position of other undetected mobiles nodes in their vicinity, creating an ad-hoc localization network. The reported localization error was 1.88 meters.

5.2 Broadcast and wide area network solutions

5.2.1 Solutions based on television signals and cellular networks

The solutions in this section are based on RF infrastructure transmitters that cover a wide area, and while not originally designed for localization, can be adapted to provide indoor localization services. In particular, two such technologies were identified: television (TV) broadcast signals [62, 63] and cellular phone networks [73].

Rabinowitz and Spilker [62, 63] proposed the use of synchronization signals already present in the Advanced Television Signal Committee (ATSC) standard for compliant digital TV signals. As the signal properties and geometrical arrangement of the TV broadcast network have been designed to penetrate indoors, they offer significantly greater indoor coverage than GPS-based solutions. Implementation of a localization solution would require no modification of the existing broadcast signal. To overcome the inherent lack of synchrony between stations due to clock imperfections, Rabinowitz and Spilker deployed a fixed reference station that transmitted an offset correction signal. A mobile station then calculated the ToF of the signal, and subsequently the distance to each TV broadcast station. As the positions of the broadcast stations were fixed and known, the position of a mobile receiver could be estimated. At least three visible transmitting stations were required for triangulation purposes. Rabinowitz and Spilker presented experiments in indoor environments in which they obtained a mean localization error ranging from 10 to 23 meters depending on the environment.

Hu et al. developed a method for the localization of mobile phones inside a cell [73] using fingerprinting techniques. Invoking a method for the automated creation of a radio fingerprint of the cellular signal, Hu et al. argued that granular localization can be achieved in indoor environments by statistically matching the fingerprint of the received signal with a record in the radio map. The authors emphasized that the localization accuracy in this case is highly dependent on the size of the cell and the characteristics of the environment. This localization solution can be considered a combination of scene analysis (the off-line phase) and proximity techniques. Unfortunately, the improvement achieved in localization accuracy over conventional cell-ID localization was not reported.

5.3 Solutions based on RFID tags

An RFID system is commonly composed of one or more reading devices that can wirelessly obtain the ID of tags present in the environment. The reader transmits a RF signal. The tags present in the environment reflect the signal, modulating it by adding a unique identification code [45, 59]. The tags can be active, i.e., powered by a battery, or passive, drawing energy from the incoming radio signal. The detection range of passive tags is therefore more limited.

Lionel et al. [45] proposed a localization system named LANDMARC, using active tags. Reference tags were located in known, fixed positions in the environment. The reader was also situated in a fixed position. To locate a mobile tag, the reader scanned through 8 different power levels for tags in the vicinity. When a mobile tag was detected, the receiver compared the power returned by the reference tags and the mobile tag, determining the closest reference tags using a nearest neighbour algorithm. The position of the mobile tag was determined by triangulating the position of the nearest reference tags. The authors reported a localization accuracy of 2 meters, 75% of the time. The maximum localization delay was of 7.5 seconds.

Jia et al. [35] proposed a hybrid radio and vision based system which used RFID tags and a stereo camera for robot navigation purposes. To estimate the location of the mobile unit, RFID tags were used to mark walls and obstacles within the environment. The RFID detector comprised of a directional antenna, which yielded the general direction of the RFID tags detected. When a tag was in proximity of the robot, it obtained images from the stereo camera to estimate the distance to the obstacle marked by the RFID tag. Subsequently, the ID of the tag was compared against entries in a tag location database to determine if it belonged to a fixed landmark (i.e. RFID tag fixed to the wall) or an obstacle prone to change its position in the environment (i.e. a chair or a human). If the tag belonged to a fixed object or location, the information from the camera, combined with the directional orientation obtained from the RFID tag, were used to estimate the distance and orientation of the robot with respect to the tag. The localization accuracy was of 8.5 centimetres. Although this localization system was designed for a robot, the constraints on the robot's movement and posture were minimal. Therefore, this system might be adapted to human localization, at the risk of accuracy reduction.

Finally, in [81], Tesoreiro et al. introduced a localization system based on proximity to RFID tags. In this system a museum visitor used a personal digital assistant (PDA), which served as an automated museum guide. To estimate the position of the user, the PDA obtained the ID of RFID tags in the vicinity. Each tag was associated with an exhibit in the museum. The ID of the detected tag was subsequently transmitted to a server, which returned information about the exhibit in proximity to the user. This information was displayed in the PDA screen. The localization accuracy of this system was not reported. However, the accuracy was related to the density of tags in the environment.

5.4 Solutions based on a radio frequency radar

Roehr et al. [71] presented an extension to the conventional frequency-modulated continuous-wave radar. The particular characteristic of this system was that it used two-way radio communication. Both fixed and mobile units were capable of transmitting and receiving a frequency modulated signal with a 5.8 GHz carrier. The fixed and mobile clocks were synchronized before distance and velocity estimations could be calculated. Once the units were synchronized, the fixed unit emitted a signal, which arrived at the mobile station. Subsequently, the mobile station sent a reply to the fixed station, which was synchronized using the signal just received from the fixed station. The round trip time of the signal was used to calculate the distance between the fixed and the mobile stations, while the frequency deviation was used to estimate the velocity of the mobile unit. The experimental setup included one experiment within an office building, where distances ranging from 5 to 25 metres were measured with the radar system. These distance measurements were then compared against the measurements obtained with a laser range finder. This experiment was designed to test line-of-sight conditions only and yielded measurements with a deviation of less than 3 centimetres when compared to laser range finder measurements.
In [65], the authors presented a radar indoor human tracking system, which exploited the Doppler effect of moving objects and micro-Doppler signal features that are particular to human movements. Various movements of the human body were classified based on the Doppler features received by the radar system. Such features were obtained from a joint time-frequency analysis, using the short-time Fourier transform and the reassigned joint timefrequency transform. The authors also proposed a scheme for tracking several human beings concurrently; the Doppler separation effect between moving humans was exploited, while target differentiation was realized using an antenna array that formed directed beams. Three antennas and two frequencies were required for multiple target tracking. Finally, the authors presented experiments to determine the range of the target, using frequency diversity, and studied the effects and errors introduced by the attenuation due to walls in the environment. The authors concluded that the properties of human movements can be exploited for localization purposes and that human localization using radar technologies is challenging within an indoor environment. They suggested a few techniques that might improve indoor localization, such as the placement of the radar as far away from the walls as possible.

5.5 Limitations

The propagation of RF signals in indoor environments poses a central challenge. Certain materials within the indoor environment affect the propagation of radio waves. For example materials such as wood or concrete attenuate RF signals, while materials such as metals or water cause reflections, scattering and diffraction of radio waves. These effects lead to multi-path radio wave propagation, which encumbers accurate calculation of the distance between the transmitter and the receiver [88, 91, 73, 42, 99, 79, 71, 52]. Several authors have proposed techniques to compensate for these inaccuracies by automatically generating radio maps which consider the structure of the building [79, 73, 34]. However, a comprehensive model of all the materials in a complex environment such as a health care facility or a patient's home is a non-trivial problem.
The propagation of radio waves are adversely affected by changes to the physical environment such as the rearrangement of furniture, structural modifications or movement

of personnel within a building. Clinical setting are under constant change; elevators, personnel and large metallic structures such as beds and wheelchairs are constantly moving through the building. In these environments, the radio properties are highly dynamic, and a radio map captured at a certain point in time cannot be used reliably for localization without accounting for these dynamic changes [45, 43, 92, 9].
Interference and noise are often-mentioned challenges [42, 84]. Although some solutions operate within a reserved radio band [62, 73, 63], most of the research is conducted on open spectrum bands. This means that these solutions must account for the increased risk of interference due to other systems sharing the same frequency bands of the radio spectrum [43].
Finally, the usage of radio transmitting devices is often times restricted in critical areas of most healthcare facilities, according to recommendations made by the Association for the Advancement of Medical Instrumentation (AAMI) [1] and other standards or regulatory bodies. These restrictions limit the deployment of localization systems based on non-broadcast radio waves to specific, non-patient care areas, e.g, waiting rooms.
Localization technologies based on RF technologies can be attractive due to the ubiquity of certain infrastructural technologies, such as wireless data networks, that may already be present in the facilities. Care must be taken, however, in evaluating the impact of the physical environment on the RF localization technologies, as the solution may be rendered non-operative in certain clinical settings.

6. Photonic energy

Light refers to the phenomena of electromagnetic radiation at wavelengths within the visible range, which extends approximately between 380 and 750 nanometres. Photonic energy refers to the energy carried by electromagnetic radiation in this wavelength range, known as visible light, or in its lower or upper vicinity, known as ultraviolet and infrared light, respectively. A photon is the minimum possible discrete amount of light energy.
Solutions in this category rely on the photonic energy received from infrared or visible light emissions or reflections, to estimate the position of an object in space.
The articles selected for this section can be distinguished based on the sensor required to estimate the location of the tracked object. Some articles proposed the use of cameras to locate a mobile subject or device via image processing [58, 5, 94, 24, 21, 93, 97, 89, 39, 70, 31, 48, 10]. In contrast, some other articles present localization solutions based on non-image processing devices [15, 56, 55, 14, 33, 90].
Methods based on image analysis are all of the scene analysis persuasion. The classical model of computer vision-based location detection consists of 4 main stages [5]

1. Image acquisition, normally through a video camera

2. Segmentation of the image and extraction of relevant features

3. Selection of the closest match or matches of the detected features against the entries of a database of features (e.g. edges or fiducials). This process typically involves mathematical transformations of the spatial relationships between the features detected to account for the variability in scale, rotation or luminance of a given scene.

4. Computation of the pose of the camera. This consists of estimating the position and orientation of the camera that could have given rise to the observed image, assumed to be a distorted and rotated version of a database image. The selection of the closest match is achieved as in the previous step.

We can subsequently distinguish between localization systems that rely on image processing that exploits natural environmental features and systems that rely on the detection of a predefined synthetic pattern (i.e. fiducial) in the environment.
Consequently, the articles included in this section were organized in three main groups: 1) image processing and natural feature extraction, 2)image processing and fiducial recognition, and 3) non-image processing sensors. This organization is presented in the following sections.

6.1 Image analysis, natural feature extraction and recognition

Natural feature extraction refers to the mathematical processing of an image to extract a set of numerical values that uniquely represent that image. Features of an image are commonly selected from its colour histogram [21] or from structural edges and their spatial relationships [5]. A reduced subset of these features is subsequently defined to uniquely identify the image.

6.1.1 Mobile camera systems

Articles grouped in this section consider solutions where the camera was carried along by the subject or device being located. In most cases the localization process involves two stages, analogous to the two-stage fingerprinting process described for wireless localization solutions. In the off-line stage, images of the environment are captured at predefined locations. Each image is processed to extract its unique features. Subsequently, the extracted features are stored in a database along with the associated camera position and orientation, at which the image was captured.
In the on-line stage, the camera captures an image, and image features are subsequently extracted. These features are compared to the entries in the feature database, either directly (to define a crude location of the camera), or through spatial transformations that yield the best match between the features in question and those in the database (to obtain more accurate location estimates). Such transformations reflect the differences in between on-line and offline image capture locations. Once the best matching set of database features is identified, the corresponding database entry for camera position and orientation is used to estimate the absolute location of the camera.
Articles adopting a mobile camera system include [5, 94, 24, 21, 93, 70, 31]. It must be noted, however, that these articles propose the indoor localization of robots. Nevertheless, the algorithms used in such articles are not strictly dependent on properties of the robots and can be easily adapted to human localization. Three distinctive examples are presented in this section.
Observing the rectilinearity of human-made indoor environments, Aider, Hoppenot and Colle [5] proposed a localization system based on a mobile monocular vision system which exploited straight line environmental features. The maximum localization error reported

during their experiments was 20 centimetres, and the maximum angle estimation error reported was 2.5 degrees.

Frontoni and Zingaretti [21] used a single colour camera to detect coloured areas of an image and their spatial relationships, which were, in turn, used as features. Through a rotation invariant feature transformation, they estimated the most probable location of the camera from a previously created database of features. The authors reported indoor localization errors less than 50 centimetres, 94% of the time.

In [31] the authors presented a navigation system for a humanoid robot, which can be extended to the case of human walking. Location determination was accomplished in two stages. In the off-line stage, images were captured while following the route between two points. The on-line stage involved autonomous robot navigation between two arbitrary points while capturing images. To achieve localization, an algorithm correlated freshly captured images with images in the route database. In this way, the unprocessed raw image was considered as a large set of features. Correlation analysis yielded the position deviation between the learned route and the current position. Temporary occlusions were detected as sudden drops in the correlation. The maximum position deviation reported during an experimental 17 metre long route, was 0.9 metres. This represents a deviation of 5.3% of the walked distance.

6.1.2 Fixed camera systems

When the camera or cameras of the system are mounted in fixed locations in the environment, the structural features of the building cannot serve as discriminant factors for localization purposes. Instead, features of the object being tracked must be used. In this way, if the salient features of the object or objects appear in the field of view of the camera, the location of the object or person can be calculated with respect to the camera's fixed position. The position of the object of interest in the environment is estimated based on its position within the captured image, and the spatial distribution of its salient features.

Although several candidate articles discussed solutions for tracking the position of a person using cameras fixed in the environment, few considered tracking a person in a building. Instead, most of the articles presented a solution in which the tracking process was limited to the field of view of a single camera. Therefore, only articles [58, 97, 89, 48, 10] were considered as an indoor tracking solution based on our inclusion and exclusion criteria.

In [58], the authors proposed a system that detected elements of the scene which were not part of the static environment. Through segmentation, the colour histograms of the region or regions of interest were obtained, along with a vertical colour average to estimate a general vertical orientation of the object detected. In this way, the algorithm accounted for the global colour scheme of the clothes of a person while standing up. A camera, located at the entrance of the test environment, was used to create an initial colour model of each person. Fifty subjects were tracked within an office environment using surveillance cameras in this fashion. Visual overlaps between cameras allowed constant tracking of subjects. The authors reported correct user recall values of 87.21% with an availability of 73.55%.

A similar approach was presented in [97]. The authors used colour-based features assuming vertical differentiation of colour regions of a human figure while standing up. To account for movement between cameras with no visual overlapping coverage, the authors proposed the creation of a connected graph that represented the areas covered by each camera. The edges of the graph denoted physical connections of the areas covered by the field of view of

each camera. This connected graph was used in the probabilistic modeling of the movement pattern and traffic constraints of the user, to improve recognition and tracking accuracy.
Pursuing a slightly different approach, [89] proposed a distributed sensor network based on image processing. The robustness of the solution relied on overlapping the visual field of several cameras, and distributing computational processing for localization purposes among the elements of the distributed network. Features were extracted using a principal components analysis of features obtained by differencing consecutive segments of the image. The field of view overlap of different cameras allowed robustness against occlusions. The authors demonstrated the feasibility of human tracking in a crowded setting. The localization accuracy, however, was not reported.
In a similar vein, [48] proposed a system with 4 cameras with partially overlapping coverage areas. Colour and non-colour features were used to account for areas of interest in the image that yielded colour and illumination information, respectively. These cameras were calibrated to estimate the 3D position of the user within the 2D image using a projective algorithm. Using an evidential filter, particles represented the probability of a user being in a certain location. Experiments were conducted with multiple users navigating a room concurrently, without deliberately avoiding occlusions. Three hundred particles were used to represent each user. In some cases the users wore similar clothes. Some of these users walked out of the field of view of some of the cameras intermittently. The reported localization accuracy of a single user using the 4 cameras was 0.15 m, while using a minimum of 30 particles for the evidential filter.
Finally, the Easyliving project [10] proposed the usage of image-based localization systems to provide context awareness within intelligent environments. The project was based on ceiling-mounted stereo vision cameras capable of estimating the anatomical posture of humans or the orientation of objects in the environment. The identification of the subject being tracked was based on colour and structural features. The localization accuracy was not reported.

6.2. Image analysis, fiducial markers

Fiducial image detection differs from natural feature detection in that the image processing algorithms are designed to detect predefined, synthetically created, patterns in the environment. These patterns are called fiducial markers [51, 19].
In principle, localization algorithms based on fiducial recognition are very similar to localization based on natural feature recognition. However, as the properties of the image to be detected are constrained, there is no need for an *a priori* stage to extract the features of the fiducial. A database is still required to determine the location of the camera relative to a reference frame, fixed with respect to the environment. In this case, however, the database can be created automatically by storing a numeric ID associated with the fiducial along with its coordinates in the environment [94]. The ID of the fiducial can be encoded within the fiducial image, analogous to how the image of a bar code represents a numeric ID. Some reported advantages of fiducial recognition over natural feature recognition are reduction of computational requirements, improved detection accuracy and resilience to noise artifacts [51].
In [94], the authors proposed a position detection system for robot navigation in indoor environments. They conducted a simple experiment using a fiducial marker as reference. They presented an image analysis technique based on homography to obtain the relative

position of the camera with respect to the fiducial marker. Although developed for a robot, the localization technique is not constrained to intrinsic properties of the robot, and can be applied to human localization.

Kim and Jun [39] introduced a wearable indoor localization system composed of a portable computer, a head mounted display (HMD) and a camera. Localization was achieved through image processing, combining fiducial markers and natural visual feature extraction. Localization with fiducials was achieved through an open source library called ARToolkit [37]. The localization algorithm detected a synthetic visual pattern in the image captured by the camera. Using affine transformations, the authors estimated the distortion and scaling of the fiducial due to angle of view and capture distance. With this information, the exact position (i.e. distance and orientation) of the camera with respect to the marker could be determined. The authors modified ARToolkit, adding an adaptive illumination thresholding algorithm and an algorithm for natural scene feature recognition, which determined the location of the person when there were no fiducial markers in view. Natural feature recognition was achieved by analyzing the color and hue histograms for a sequence of frames. The authors defined a "location" as a sequence 64 consecutive frames. Hence, each location was defined by a high dimensional space, namely a sequence of 64 colour and 64 hue histograms. A Linear Discriminant Analysis (LDA) was applied to reduce the dimensionality of the features which defined each location. The first five LDA coefficients of each frame were used as descriptive features of that frame. The sequence of features in consecutive frames defined a location. An off-line navigation phase facilitated the creation of a database of natural features of the building. To estimate the location of the user, the vector of features obtained from the last 64 frames, defining the current location, were compared against the features in each entry of the database. The difference between a feature vector captured by the camera in the on-line stage against feature vectors stored in the database were quantified by the Euclidean distance. If the Euclidean distance was lower than an unspecified threshold, the location of the person was defined as the matching database coordinates. In this fashion, the system yielded the general location of the user in a continuous fashion. The exact location of the user was determined using fiducial markers in the field of view of the camera. The location information obtained by both visual systems was presented in a virtual map, along with instructions to complete a predefined path through the HMD. The localization accuracy of the system was not reported.

6.3 Other photonic sensors

Articles included in this section make use of non-image capturing Infrared (IR) sensors [15, 56, 55, 14, 33, 90].

In [15], the authors presented an IR proximity-based localization system, which provided museum visitors with useful information about exhibits in each hall. For this purpose, IR emitters were installed in the ceiling of the door frames of every room. Each emitter transmitted a unique ID using the Infrared Data Association (IrDA) protocol. The visitor carried a Personal Digital Assistant (PDA) with an infrared port. The PDA contained a database of visual and textual information of the exhibits, as well as maps of the museum. Upon reception of a new ID, the PDA automatically presented the map of the corresponding hall. While in the hall, a graphical user interface in the PDA helped the visitor to obtain information about a particular exhibit. The authors noted some problems while deploying this localization system, in particular, noise and reflections of IR signals. The user localization granularity of this system was down to the scale of a room.

A combination of scene-analysis and triangulation is presented in [56]. In this solution, a unique ID was modulated by the IR emitter. The carrier frequency used for modulation was changed in a cyclic way, from low to high frequencies. As the attenuation properties of an infrared signal are frequency-dependent, ID's modulated at lower frequencies can be successfully detected farther away from the emitter. However, the power of the IR signal decays in a nonlinear fashion with distance. To characterize this power decay, the authors obtained off-line measurements of the received signal in the vicinity of the emitter, in a similar fashion to fingerprinting for RF signals. The authors measured the ID detection success rate in 10 cm concentric regions, at steps of 5 degrees, repeating this process for different modulating frequencies. Instead of creating a database of the signal in each point, the authors modeled the decay of the signal with an equation that was dependent of the orientation of the receiver, the distance between receiver and emitter, and the modulation frequency. Using this approach the system achieved a maximum localization error of 10 centimetres, within an area of 5 square metres. Although a building-wide experiment was not conducted, the usage of a unique ID per transmitter would allow the deployment of multiple emitters in a cellular arrangement to cover large areas.

Petrellis et al. [55] presented a localization system consisting of two IR emitters fixed in the environment, and two receivers installed on a mobile unit. The transmitters were mounted facing each other on the walls of a corridor, while the receivers on the mobile unit faced away from each other. Each emitter transmitted a series of unique cyclic data patterns, modulated at a carrier frequency of 38 Hz. The opposite-facing arrangement of the receiving sensors facilitated detection of user orientation and, at the same time, discrimination between signals received via direct path and reflected signals. Reflection rejection was enhanced using a predictive model, which took into consideration the immediate previous location and orientation of the user. Since the system relied on predictions of future position and orientation of the mobile device, the estimation rules constrained the maximum linear and angular speeds at which the system was reliable. Given that the emitters transmitted unique sequences, the authors proposed a cellular-like spatial emitter arrangement to cover extensive areas. The system was sensitive to moving personnel and other objects that caused reflections although compensation algorithms reduced such effects. The localization accuracy was not reported.

Cheok and Li [14] presented a localization system which made use of existing fluorescent lamps in a building. A user carried a wearable computer instrumented with a photo-detector and a gyroscope. Each fluorescent lamp emitted an ID, associated with the location of the lamp. The ID was encoded through pulse frequency modulation. The encoded information did not introduce perceptible illumination effects. The photo-detector and gyroscope were mounted on a cap, worn by the user. The gyroscope served to obtain orientation information. The wearable computer was carried on a vest, worn by the user. The area illuminated by two lamps transmitting different location ID's could not overlap, since this would cause interference, resulting in the inability to detect the encoded signal. Whenever available, location information was presented to the user via the wearable computer. Such information was overlaid on the user's visual field through a HMD. The reported localization accuracy was within three to four meters. Although this accuracy is dependent on the lamp and user heights, these numbers were not provided. The minimal separation between lamps was of 2.31 metres.

iGPS, commercialized by Metris (formerly ArcSecond) [33], is a triangulation-based localization system for tracking assets, personnel or any other mobile elements. To accurately estimate the position of a receiver, a pair of eye-safe IR laser emitters radiate a signal using two different wavelengths, while a an infrared strobe provides a reference signal. Using signals from 3 or more transmitters, the receiver calculates and transmits its position to a central data collecting station. In order to estimate the orientation of a solid body, two or more receivers are attached to it. The iGPS system claims to offer sub-centimetre accuracies.

Scientists from Olivetti Research designed the Active Badge location system, which consisted of small badges that transmitted a unique ID via IR emitters [90]. The badges were worn by people, who could then be located when in the vicinity of a receiving station. The badge transmitted a unique identification code every 10 seconds. The system offered sub-room accuracy, and was used to redirect phone calls for personnel.

6.4 Limitations

Common problems reported for photonic sensor localization are the ambient noise in the form of light or thermal radiation [5, 56, 97], signal reflections [15, 44, 56] and in the case of image processing solutions, illumination variability [58, 89, 78, 97].

In image processing solutions, ambient noise is usually overcome by image filtering techniques. In the case of IR sensing, effects of ambient noise can be mitigated by using a combination of different modulation frequencies [56, 33].

Another problem commonly mentioned in the surveyed articles is the occlusions caused by dynamic elements of the environment [5, 24, 21, 89, 10]. For instance, in the experiments performed in [56], the introduction of new objects or humans was specifically avoided during the experiments. In order to reduce the risk of visual occlusions by humans and objects, solutions comprising building-mounted equipment commonly install the detection equipment on the ceiling [15, 90]. Another way of reducing occlusion is to deploy sensors with overlapping coverage areas [58, 89, 33].

However, clinical settings and public indoor areas like shopping malls are oftentimes densely populated. Consequently, occlusion conditions can emerge frequently even with ceiling mounted sensors. Therefore, if a photonic-based system is to be considered for localization purposes, it may be advantageous to simultaneously invoke a secondary tracking system to assist in the localization process during periods of optical occlusion.

In the case of laser based-solutions, only class 1 laser devices should be used, which are classified as "eye-safe" by the IEC 60825-1 standard [32]. In clinical settings, however, special care must be taken with even class 1 laser devices, to ensure that no harm will be caused by concentrated light on the skin or eyes of light-sensitive patients.

Finally, we should emphasize the privacy issues that may arise as secondary to a localization solution. Health care facilities operate under the precepts of information non-disclosure to protect the privacy of personnel, patients and clients. This is an important consideration when a localization system is designed to capture images of the environment, as such images can reveal important information about the person wearing the system or about the patients and health care personnel in the vicinity. Frequently, image processing mobile localization devices are designed to send the captured image to central, computationally powerful servers for image processing. The confidentiality of the image is at great risk of being compromised while in transit over a network. Therefore, we strongly recommend against the transmission of raw images through wireless data networks. Instead, the position must be estimated by the image capturing device, and the image must be discarded immediately to minimize risk of privacy breach.

7. Localization detection based on sonic waves

Sonic waves are mechanical vibrations transmitted over a solid, liquid or gaseous medium. The distance traveled by a sonic wave can be indirectly calculated by exploiting the quasi-constant speed of such waves in air. Sonic waves produced by vibrations below and above the threshold of human hearing are known as infrasonic and ultrasonic waves, respectively.

The localization solutions grouped in this section propose the usage of ultrasonic range finders and sonars. All the articles presented herein use triangulation-based localization techniques based on the time of flight (ToF) of a sonic wave in air [27, 12, 13, 80, 84, 85, 68, 100, 60, 2, 26].

Most of the solutions in this section use a hybrid technology approach, exploiting the difference in propagation speeds of RF and ultrasonic waves [12, 13, 80, 68, 100, 60, 2]. Localization is achieved by measuring the ToF of ultrasonic waves for triangulation purposes. To achieve high localization accuracy, the transmitter and the receiver must be synchronized. An RF wave travels several orders of magnitude faster than a sonic wave. Thus, when a combination of these two types of signals is emitted in unison, the difference between the time of arrival (ToA) of the radio and sonic waves at the receiver side is a good approximation of the ToF of the sonic wave. Therefore RF waves are used for synchronization purposes, while ultrasound waves are used for triangulation purposes. When the RF synchronization signal is transmitted by a bluetooth device, the system is commonly known as BLUPS, which stands for "Bluetooth and Ultrasound Positioning System". Two BLUPS have been included in this survey [12, 13]. We continue by presenting two distinctive examples of RF/ultrasound hybrid solutions.

Teller, Jiawen and Balakrishnan [80] extend the localization system called 'Cricket' [60] by making it pose-aware. The authors combined several ultrasonic receivers in a single mobile unit. The separation between each sensor in the mobile unit was fixed. The phase differences between the signal received at each sensor of the array were used to calculate the orientation of the device with respect to the transmitter. This emulates the human process of detecting the direction of an incoming sound. This extended version of Cricket located the mobile unit with sub-centimetre accuracy, and reported angle accuracies were down to a few degrees.

The localization system proposed in [68] is based on a technology called 'the Bat' [2, 26]. In the Bat, the ultrasound signal is emitted by the mobile device. The ceiling of the environment is instrumented with several RF transmitter-ultrasound receiver units. The ceiling mounted units were interconnected and synchronized. After simultaneously emitting a radio pulse, the ceiling units waited for a reply from the mobile stations.

A mobile unit in turn sent a reply as soon as it detected a radio beacon. The creators of the Bat reported accuracies under 9 centimetres, 95% of the time.

In a different vein, three of the selected articles presented localization systems based solely on ultrasound waves [27, 84, 85]. In these solutions, wide band sonic signals were required.

To achieve accurate localization, spread spectrum code division techniques were used. Spread spectrum code division allows several emitters to transmit signals, sharing the same frequency band concurrently, causing minimal interference to other signals being transmitted [54].

Hazas and Hopper [27] presented a localization system called "Dolphin" which employs broadband ultrasonic waves. The advantage of broadband ultrasound is the ability to localize a mobile station with ultrasound signals without an external synchronization system. In particular, Direct Sequence Code Division Multiplexing (DS/CDM) techniques were used to combine multiple signals simultaneously over the same frequency spectrum. The authors reported maximum localization errors of 10 centimetres 95% of the time. Along the same lines, in [85] broadband ultrasound signals and DS/CDM techniques were used for localization purposes. Beacon units were fixed at predefined positions of the environment. Such beacons transmitted a unique ID using DS/CDM. The receiver location was calculated via hyperbolic trilateration, using the difference between the time of arrival difference of time of arrival (DToA) of the signals received from the beacon. The authors reported localization errors in the millimetre range. Although the localization system was proposed for utilization on the scale of a full building, experiments were only conducted in a limited area.

7.1 Limitations

In this section we list the common challenges associated with sonic wave localization systems. Some authors reported that high levels of ambient noise commonly encumber the detection of the sonic signal [27, 13, 82]. This is particularly important when considering the deployment of a sonic localization solution for healthcare facilities, where areas of the building can be densely populated and noisy (e.g., emergency rooms or hospital foyers).

Another common issue is the co-interference caused by the presence of multiple sonic emitters in the environment. This condition encumbers the isolation of single sources. Common narrow-band ultrasound emitters may be affected by this condition [27]. The most common emitter disambiguation technique is to combine ultrasound-based triangulation with RF beacons. These composite solutions communicate the emitter ID via a different physical channel, assign time slots to multiple emitters, and thereby avoid interference at the cost of reduced accuracy.

Recent research proposes broadband ultrasonic emitters as means of overcoming the concurrent interference problem associated with narrow-band sonic sensors. The usage of a wideband signal allows for multiple access techniques such as DS/CDM, which are commonly used in telecommunication solutions (i.e., cellular phone networks. These techniques provide improved noise resilience, while allowing multiple emitting stations to transmit in synchrony over a single ultrasound channel. This eliminates the need for additional communication channels [27], reducing the complexity of the system.

All the solutions considered in this survey estimate the position of a mobile device based on the triangulation of the ToF of a sonic wave. The speed of sound over air is an important factor in such calculations. Temperature variations are known to affect the speed of sound in air [27, 12, 13, 82, 26]. Therefore, sonic wave based systems cannot be used in environments with frequent and drastic temperature or environmental changes [13].

Finally, the propagation properties of sonic waves in indoor environments pose a challenge for accurate position estimation. Elements in the environment such as furniture, walls and their salient edges cause echoes. The appearance of such echoes can lead to localization inaccuracies [13, 84, 68, 26]. Obstructions between the receiver and the transmitter can cause NLOS conditions, which contribute to erroneous distance estimates [12, 13]. The dynamic nature of healthcare facilities represents a challenge. Installation of ceiling-mounted transmitters or receivers may, to a certain extent, alleviate some of the problems associated with environmental conditions.

8. Localization detection based on inertial or mechanical sensors

The articles included in this section measure energy exerted by the mechanical movement of the element being tracked. Such energy can be measured via direct application of force, or by exploiting the inertial properties of an element of negligible mass (when compared to the mass of the object being tracked) that deflects from its fixed position within a reference frame when it is subject to acceleration or angular rotation.
Three articles were included in this section, one based on mechanical contact [50], and two based on inertial sensing [17, 66].

8.1 Localization detection based on mechanical coupling or activation

A sensory block consisting of a 60 by 60 centimetre metallic shelf was instrumented with load cells. Several sensory blocks were used as the support structure for a wooden floor. The separation between sensory blocks was of 20 centimetres. An experiment for tracking a single user was conducted, reporting a localization accuracy of 28.3 centimetres 85% of the time. Experiments were designed to track two users with intersecting paths following different walking patterns. The introduction of a second user reduced the localization accuracy during experiments involving non-intersecting paths, to 28.3 centimetres 76% of the time. The system could only differentiate between users if their weights were extremely different, i.e. the experiments were conducted with two participants, weighing 50 and 90 kilograms. Accuracy was not reported.
Similarly, in a project called "Smart Floor" [50], metallic plates were instrumented with load cells. These plates were then laid on the floor. In order to identify the person walking over a plate, the signal captured via the load cell was processed in order to select a set of 10 features. Such features emerged as distinctive for each pedestrian. The system required an off-line stage, in which the users to be identified walked over the plates of the Smart Floor. The data captured during the off-line stage served to create a database of stepping features for each user. Later, during the on-line stage, the features extracted via the load cells are matched with the features stored in the database, using a nearest-neighbour algorithm. 15 participants were tracked and identified during an experiment. The system achieved a user recognition rate of 93%. As well, the authors investigated the effects of different footwear on recognition accuracy, concluding that there as no effect. Since Smart Floor relies on mechanical contact, it can be classified as a proximity-based localization system.

8.2 Localization detection based on inertial sensors

All the articles included in this section considered a hybrid solution, combining inertial sensors with a different localization technology that provided absolute positioning information [17, 66, 14]. Since inertial sensors yield relative positioning information only, an absolute reference is required to specify the displacement reported by an inertial measurement in absolute coordinates. Inertial sensors proposed in the selected articles include gyroscopes, accelerometers and inclinometers.

Evennou and Marx [17] proposed a localization system based on an IEEE 802.11 wireless network. The absolute positional information obtained from the wireless network was combined with the relative displacements and rotations reported by a gyroscope, a dual-axis accelerometer and a pressure sensor. The information obtained from all the sensors was combined through Kalman and particle filtering [76, 8]. To calculate the displacement of the user, the accelerometer was used to count the number of steps taken. Then, a constant estimate of the stride length of the user was used to calculate the total displacement of the user within the environment. The authors showed that combining the information from the bank of sensors yielded improved localization when compared to the usage of each sensor separately. Eve-nou and Marx conducted an experiment using their multi-sensor localization system. The localization accuracies reported during such experiment ranged from 1.53 to 3.32 metres.

In [66], Retscher presented another multi-sensor localization system, which included iner-tial sensors. This system was based on a localization infrastructure named "ipos". The system was composed of a combination of a fingerprint-based Wireless Local Area Network (WLAN) localization technology, a digital compass, a pressure sensor, and three accelerometers, plus a GPS unit for outdoor localization. Retscher characterized different brands and makes of each type of sensor. By analyzing the inertial sensors separately, localization accuracies between 5 to 8 metres were obtained. The distance traveled by the user was calculated based on stride frequency, detected through accelerometry. A pre-estimate of the stride length was used to estimate the displacement of the user. The system was reported as highly dependent on the walking patterns of each user. Finally, Retscher conducted an experiment in an indoor environment, using the wireless network localization system combined with the dead reckoning sensors, reporting approximate localization accuracies of 3 metres. Kalman filtering was invoked to combine the inputs from multiple sensors.

The solutions based on inertial sensing were classified as dead reckoning localization techniques, since the location estimates provided by such sensors depend on previous measurements to estimate the absolute position or orientation of the object being tracked at a given instant.

8.3 Limitations

One of the main issues with inertial sensors is the drift associated with thermal changes and inherent noise [17].

Drift measurement deviations are mainly caused by thermal changes in the circuitry of the sensor. The effects of such deviations can significantly affect the location estimation process. Since double integration is required to estimate the displacement of an object based on its acceleration, any small measurement error will be accumulated over time, leading to aberrant position estimation. To avoid this condition, in [17] and [66] the displacement of a user is estimated considering a constant stride length. In this case, however, the accuracy of the system is highly dependent on the walking pattern of the user.

As mentioned earlier, inertial sensors can only yield relative motion estimates. Therefore, a system that provides an absolute positional reference is required to report absolute location estimates. While an absolute positioning system could be used as the unique localization technology, the addition of the information provided by inertial sensors can lead to improved localization accuracies, as discussed earlier. Recall that solutions based on photonic and sonic sensors can be rendered unavailable due to occlusions. A dead reckoning system could be used to account for such periods of unavailability.

9. Other localization technologies

This section groups a reduced number of articles that were not included in any of the previous sections, but which matched the inclusion criteria. The sensing technologies included in this section were pressure and magnetic sensors.

Evennou and Marx [17] and Retscher (2007) [66] used atmospheric sensors as part of a multi-sensor localization system to estimate the altitude of a user. In both articles, vertical storey level accuracy is reported. In [66], Retscher characterized different commercial pressure sensors, recommending the pressure sensor PTB 220 manufactured by Vaisala. This sensor yielded an accuracy of 33 cm.

In [66], a digital compass was used in a multi-sensor localization system to assist with bearing estimation. The author reported spurious electromagnetic field disturbances that affected the readings of the compass when in proximity to metallic structures or radio wave emitting devices.

10. Discussion

A brief summary of localization principle, merits and limitations of the aforementioned technologies is presented in Table 1.

Main technology	**Technology details**	**Localization technique t**				**Disadvantages**	**Advantages**	**Position**	**Orientation**
		TR	**PR**	**SA**	**DR**				
RF	Wireless Presonal and local area networks	✓	✓	✓		Coarse localization, oftentimes requires an offline phase. Sensitive to interference, signal propagation effects, and dynamic environmental changes	Usage of readily deployed equipment, reduced cost	Yes, down to a few metres	Yes, coarse: i.e. 4 orientation options
	Broadcast networks	✓		✓		For cell phones, a radio map of received power is required. For solutions based on TV broadcasts, reference stations are required	Usage of readily available infrastructure	Yes, down to tens of metres	No
	RFID		✓	✓		Limited localization accuracy. Limited range with passive tags. Battery replacement using active tags	Low tag cost, active tags are more expensive and require a battery	Yes, down to a few metres	N/A
	Radar	✓				Only works with line of sight. Sensitive to reflections	Good accuracy	Yes, Submeter level	No

Photonic	Image processing -natural feature extraction			✓		High processing requirements, dependence on illumination conditions and environmental noise. Sensitive to obstructions and dynamic environmental changes	High localization and orientation accuracies	Yes, submeter level	Yes, high accuracy when the cammera is mobile
	Image processing - fiducial markers			✓		Deployment of fiducials requirement & measurement of their exact positions, sensitive to obstructions	Lower processing requirements when compared to natural feature extraction	Yes, submeter level	Yes, high accuracy
	Non-image processing based	✓	✓	✓		Sensitivity to ambient noise and obstructions. Sensitivity to reflections in some cases. Affected by dynamic environmental changes	Simplicity, light weight and low cost	From room level down to submeter level	Yes, (only for one of the articles reviewed
Sonic, Ultrasonic		✓				Most solutions require external synchronization (RF beacons). Affected by ambient noise. Accuracy affected by propagation issues and NLOS. Co-interference when using narrow band emitters. Speed of sound variations, dependent on temperature and other environmental conditions	Exxtremely high localization accuracies	Yes, down to a few centimetr es	Yes
Inertial / Mechanical			✓		✓	Drift inherent to sensors. Relative localization, requirement of initialization and calibration	Self-containme nt. Resilience to environmental conditions. Continuous update of location estimates	Yes, accelerom eters, accuracy dependen t on recalibrati on frequency	Yes, gyroscopes, high accuracy, dependent on recalibration frequency
Other	Pressure sensors				✓		provides positioning information in vertical axis	Yes, vertical, submeter level	No
	Digital compass				✓	Strongly affected by electromagnetic fields and metallic objects in the vicinity of the sensor, therefore not highly reliable in indoor environments	self containment	No	Yes

Table 1. Summary of the localization technologies. The reported localization accuracies are based on typical literature values, where available. For hybrid solutions, accuracies are reported for the main technology only.
(† TR = Triangulation, PR = Proximity, SA = Scene Analysis, DR = Dead Reckoning)

10.1 Recommendations

The foregoing review has suggested 1) a need for research towards rehabilitation of topographical disorientation, 2) the existence of devices for ameliorating cognitive impairments, but limited research and development of devices suitable for topographical disorientation, and 3) the existence of research towards the development of localization systems that could be incorporated into assistive technologies for topographical disorientation. In this section, we will present some recommendations to guide the selection of a suitable localization technology for assisted navigation.

Based on previous research, each type of localization system has advantages and shortcomings. By combining some of these technologies, we may exploit the advantages of the individual technologies while mitigating their respective shortcomings. Some of the selected articles in the last section have proposed effective combinations of technologies [17, 66] or mention the advantages of such composite systems [13, 5, 94, 91]. We will briefly consider the types of technologies that may be especially conducive to combination.

Recall that one of the main shortcomings of technologies based on scene scanning and triangulation techniques for localization purposes is the dynamic nature of indoor environments. Changes in the environment will diminish the efficacy of the localization algorithms used in the system. For instance, furniture or personnel agglomeration in an office could adversely affect both laser range scanners and RF systems, leading to temporary or permanent failures in localization detection. On the other hand, the self contention of dead reckoning systems allows them to provide constant tracking of the desired target because their measurements do not depend on external elements. This self contention also means that solutions based on dead reckoning will normally require calibration and acquisition of initial position and altitude. Finally, the self-contained nature of dead reckoning systems makes them prone to drifting errors, corrupting the localization information delivered by the sensors from true values [17].

Therefore the combination of a system based on scene scanning or triangulation localization techniques with a system using dead reckoning sensors would provide the strengths of both systems, allowing real time tracking of the element to be located provided by the dead reckoning sensors, even if the other localization system is unavailable due to environmental factors, while the scene scanning or triangulation-based system would provide accurate position information for recalibration purposes and a start point for initial measurements.

In fact, such a combination is present in the articles surveyed in the last section [17, 66]. The selection of technologies to be used will be dependent on the requirements imposed by the application. For instance, a solution based on RF signals would not be desirable in environments with high RF noise, or the solution proposed would conflict with elements present in the environment; for instance IEEE 802.11 access points should not be installed in the same building as medical equipment using the same frequency band reserved for medical equipment (ISM), or localization systems based on sonic waves should not be used in environments with unstable temperature conditions. Other constraints should take into account the complexity and cost of installation of a particular solution.

Certain technical challenges must be taken into account while considering the combination of multiple types of sensors to create a robust localization system. Since the localization update rates and errors associated to each type of sensor can be different, optimal schemes to combine localization data from multiple error-prone sources must be used. We can see that in fact, several algorithms of this sort are mentioned in the articles reviewed, for instance, Kalman filtering and particle filtering among other techniques. One more aspect to consider is that, the addition of elements to a system normally increase the complexity of such system, making it more prone to unavailability conditions due to failure of one or more components.

Other constraints to be taken into account are the complexity and cost of deployment of a particular solution. For instance, for some solutions the pre-existing deployment of infrastructure might reduce time and cost of implementation of a localization system. For instance, some solutions use existing unmodified IEEE 802.11 infrastructure networks [98, 92, 17, 43, 66, 42, 53, 79, 57, 9], and some others benefit from already existing video surveyance systems [58, 97].

11. References

[1] AAMI. Technical information report (tir) 18, guidance on electromagnetic compatibility of medical devices for clinical/biomedical engineers. Technical Information Report TIR-18, Association for the Advancement of Medical Instrumentation, Arlington, Virginia, 1997.

[2] Mike Addlesee, Rupert Curwen, Steve Hodges, Joe Newman, Pete Steggles, Andy Ward, and Andy Hopper. Implementing a sentient computing system. *Computer,* 34(8):50-56, 2001.

[3] GK. Aguirre and M. D'Esposito. Topographical disorientation: a synthesis and taxonomy. *Brain,* 122 (Pt 9):1613-28, Sep 1999.

[4] Uzair Ahmad, Andrey V. Gavrilov, Sungyoung Lee, and Young-Koo Lee. A modular classification model for received signal strength based location systems. *Neurocomputing,* 71(13-15):2657 - 2669, 2008. Attractive methods;Classification methods;Classification models;Environmental factors;Feature spaces;Indoor positioning systems;Location ac-curacies;Location estimation;Location estimations;Location systems;Modular classi-fiers;Multilayer perceptron;Null values;Radio signals;Received signal strength;Received signal strengths;Target sites;.

[5] Omar Ait Aider, Philippe Hoppenot, and Etienne Colle. A model-based method for indoor mobile robot localization using monocular vision and straight-line correspondences. *Robotics and Autonomous Systems, v 52, n 2-3, Aug 31, 2005, p 229-246,* 2005.

[6] Murat Alemdar, Pervin Iseri, Irem Yalug, Huumlseyin Kutlu, Husnu Efendi, and Sezer Sener Komsuoglu. Route learning impairment associated with encephalomalasia secondary to traumatic brain injury: A case report. *Applied Neuropsychology,* 15(2):150-155, April 2008.

[7] JM Alisky. Integrated electronic monitoring systems could revolutionize care for patients with cognitive impairments. *Medical Hypotheses,* 66(6):1161-1164, 2006.

[8] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. In *IEEE transactions in Signal Processing,* volume 50, pages 174-188, 2002.

[9] Paramvir Bahl and Venkata N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *INFOCOM (2),* pages 775-784, 2000.

[10] Barry Brumitt, Brian Meyers, John Krumm, Amanda Kern, and Steven A. Shafer. Easyliving: Technologies for intelligent environments. In *HUC,* pages 12-29, 2000.

[11] R. Brunsdon, L. Nickels, M. Coltheart, and P. Joy. Assessment and treatment of childhood topographical disorientation: A case study. *Neuropsychol Rehabil,* 17(1):53-94, 2007.

[12] R. Casas, A. Marco, J.J. Guerrero, and J. Falco. Robust estimator for non-line-of-sight error mitigation in indoor localization. *Eurasip Journal on Applied Signal Processing, v 2006, 2006, p 43429,* 2006.

[13] Roberto Casas, David Cuartielles, Alvaro Marco, Hector J. Gracia, and Jorge L. Falco. Hidden issues in deploying an indoor location system. *IEEE Pervasive Computing, v 6, n 2, April/June, 2007, p 62-69,* 2007.

[14] Adrian David Cheok and Yue Li. Ubiquitous interaction with positioning and navigation using a novel light sensor-based information transmission system. *Personal and Ubiquitous Computing,* 12(6):445 - 458, 2008. Electronic ballasts;Fluorescent light;GPS tracking;Guidance and navigation;Indoor environments;Indoor positioning;Information transmission;Light sensors;Modulation techniques;Positioning information;Transmitter circuits;.

[15] C. Ciavarella and F. Paterno. The design of a handheld, location-aware guide for indoor environments. *Personal and Ubiquitous Computing, v 8, n 2, 2004, 82-91,* 2004.

[16] S.S. Dikmen, J.E. Machamer, J.M. Powell, and N.R. Temkin. Outcome 3 to 5 years after moderate to severe traumatic brain injury. *Arch Phys Med Rehabil,* 84(10):1449-1457, 2003.

[17] F. Evennou and F. Marx. Advanced integration of wifi and inertial navigation systems for indoor mobile positioning. *EURASIP Journal on Applied Signal Processing, v 2006, n 17, 2006, 11 pp.,* 2006.

[18] Shih-Hau Fang and Tsung-Nan Lin. Indoor location system based on discriminant-adaptive neural network in ieee 802.11 environments. *IEEE Transactions on Neural Networks,* 19(11):1973 - 1978, 2008. Access points;Adaptive;Adaptive neural networks;Discriminative components;IEEE 802.11;Indoor location sys-tems;Localization algorithms;Localization systems;Location fingerprinting;Multilayer perceptron;Nearest neighbors;Network learnings;Nonlinear relationships;Received signal strengths;Redundant informations;Test beds;Wireless lans;.

[19] Mark Fiala. Artag, a fiducial marker system using digital techniques. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVFR'05) - Volume 2,* pages 590-596, Washington, DC, USA, 2005. IEEE Computer Society.

[20] F. Forno, G. Malnati, and G. Portelli. Design and implementation of a bluetooth ad hoc network for indoor positioning. *IEE Proceedings: Software, v 152, n 5, October, 2005, Microsoft Research Embedded Systems, p 223-228,* 2007.

[21] Emanuele Frontoni and Primo Zingaretti. An efficient similarity metric for omnidirectional vision sensors. *Robotics and Autonomous Systems, v 54, n 9, Sep 30, 2006, p 750-757,* 2006.

[22] Chunwang Gao, Zhen Yu, Yawen Wei, Steve Russell, and Yong Guan. A statistical indoor localization method for supporting location-based access control. *Mobile Networks and Applications,* 14(2):253 - 263, 2009. Confidence intervals;Estimation errors;Indoor environments;Locally weighted regressions;Location awareness;Location-based access control;Maximum-likelihood estimations;Measured signals;Off-line trainings;On-line es-timations;Scatter plots;Signal strengths;Statistical indoor localization;Wireless signal strength;.

[23] Camillo Gentile, A. Judson Braga, and Alfred Kik. A comprehensive evaluation of joint range and angle estimation in indoor ultrawideband location systems. *Eurasip Journal on Wireless Communications and Networking,* 2008, 2008. Angle estimations;Angular resolutions;Bearing capabilities;Bearing systems;Center frequen-cies;Channel sounders;Comprehensive evaluations;First arrivals;Higher frequencies;Line of sights;Location systems;Multi paths;Non line of sights;Physical limits;Range er-rors;Ranging systems;Spatial temporals;Time resolutions;Transmitter and receivers;Uwb systems;Vector network analyzers;Virtual antenna arrays;Wall materials;.

[24] C. Giovannangeli, P. Gaussier, and J.P. Banquet. Robustness of visual place cells in dynamic indoor and outdoor environment. *International Journal of Advanced Robotic Systems, v 3, n 2, June, 2006, p 115-124,* 2006.

[25] M. Grigoras, O. Feiermann, and U.D. Hanebeck. Data-driven modeling of signal strength distributions for localization in cellular radio networks. *Automatisierungstechnik, v 53, n 7, July 2005, 314-21,* 2005.

[26] Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, and Paul Webster. The anatomy of a context-aware application. In *Mobile Computing and Networking,* pages 59-68, 1999.

[27] Mike Hazas and Andy Hopper. Broadband ultrasonic location systems for improved indoor positioning. *IEEE Transactions on Mobile Computing, v 5, n 5, September/October, 2006, p 536-547,* 2006.

[28] Mohammad Heidari and Kaveh Pahlavan. Identification of the absence of direct path in toa-based indoor localization systems. *International Journal of Wireless Information Networks,* 15(3-4):117 - 127, 2008. Angle of arrivals;Detected Direct paths;Direct paths;Do-mains;High precisions;Hypothesis testing;Indoor positioning;Indoor positioning systems;Localization errors;Multipath components;NLoS identification and mitiga-tion;On times;Ranging measurements;Received signal strengths;Receiver locations;ToA-based indoor localization;Undetected direct paths;UWB communication;Wide bands;.

[29] Jeffrey Hightower and Gaetano Borriello. Location systems for ubiquitous computing. *IEEE Computer,* 34(8):57-66, August 2001.

[30] Tao Hu, G. Lachapelle, and R. Klukas. Controlled gps signal simulation for indoors. *Journal of Navigation, v 60, n 2, May 2007, 265-80,* 2007.

[31] Junichi Ido, Yoshinao Shimizu, Yoshio Matsumoto, and Tsukasa Ogasawara. Indoor navigation for a humanoid robot using a view sequence. *International Journal of Robotics Research,* 28(2):315 - 325, 2009. Human environments;Humanoid;Humanoid robots;Image blurs;Indoor navigations;Localization and navigations;Motion capture sys-tems;Navigation functions;Quantitative effects;View sequence;View-based approaches;.

[32] IEC. Safety of laser products - part 1: Equipment classification and requirements. Technical Information Report 2nd edition, International Electrotechnical Commission, 2007.

[33] Arc Second Inc. Constellation-3d indoor gps technology for metrology. In *White paper 071502*, page 111, 2002.

[34] Yiming Ji, Saad Biaz, Santosh Pandey, and Prathima Agrawal. Ariadne: a dynamic indoor signal map construction and localization system. In *MobiSys '06: Proceedings ofthe 4th international conference on Mobile systems, applications and services*, pages 151-164, New York, NY, USA, 2006. ACM.

[35] Songmin Jia, Jinbuo Sheng, Erzhe Shang, and Kunikatsu Takase. Robot localization in indoor environments using radio frequency identification technology and stereo vision. *Advanced Robotics*, 22(2-3):279 - 297, 2008. Localization;Obstacle detection;Stereo camera;.

[36] Wei-Wen Kao and Chin-Lang Tsai. Carrier phase indoor positioning using pseudolites and relative movements. *Journal ofthe Chinese Institute ofEngineers, Transactions of the Chinese Institute of Engineers,Series A/Chung-kuo Kung Ch'eng Hsuch K'an, v 28, n 5, September, 2005, p 899-903*, 2005.

[37] Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings of the 2nd International Workshop on Augmented Reality (IWAR 99)*, pages 85-94, San Francisco, CA, USA, October 1999.

[38] Changdon Kee, Doohee Yun, and Haeyoung Jun. Precise calibration method of pseudolite positions in indoor navigation systems. *Computers and Mathematics with Applications, v 46, n 10, November/December, 2003, p 1711-1724*, 2003.

[39] JongBae Kim and HeeSung Jun. Vision-based location positioning using augmented reality for indoor navigation. *IEEE Transactions on Consumer Electronics*, 54(3):954 - 962, 2008. Accurate;Augmented realities;Bus services;Consumer applications;Display locations;Efficient;Guide systems;Image databases;Image sequence matching;Image se-quences;Indoor environments;Indoor navigation;Indoor navigations;Location informa-tions;Location model;Location models;Location positioning;Location recognitions;Plate systems;Prior knowledges;Shopping assistances;Vision-based;.

[40] C. Krause, H. Schmitz, and M. Kuipers. Predicting the availability of (a)gps in indoor places by simulations. *ITG-Fachbericht, n 194, 2006, 109-13*, 2006.

[41] O. Krejcar and J. Cernohorsky. Wifi user localization for intelligent crisis management predictive data push technology framework. *WSEAS Transactions on Environment and Development, v 1, n 2, Nov. 2005, 319-25*, 2005.

[42] Azadeh Kushki, Konstantinos N. Plataniotis, and Anastasios N. Venetsanopoulos. Kernel-based positioning in wireless local area networks. *IEEE Transactions on Mobile Computing, v 6, n 6, June, 2007, p 689-705*, 2007.

[43] Andrew M. Ladd, Kostas E. Bekris, Algis P. Rudys, Dan S. Wallach, and Lydia E. Kavraki. On the feasibility of using wireless ethernet for indoor localization. *IEEE Transactions on Robotics and Automation, v 20, n 3, June, 2004, P 555-559*, 2004.

[44] Suk Lee, Kyoung Nam Ha, and Kyung Chang Lee. A pyroelectric infrared sensor-based indoor location-aware system for the smart home. *IEEE Transactions on Consumer Electronics, v 52, n 4, November, 2006, p 1311-1317*, 2006.

[45] M.Ni Lionel, Yunhao Liu, Yiu Cho Lau, and Abhishek P. Patil. Landmarc: Indoor location sensing using active rfid. *Wireless Networks, v 10, n 6, November, 2004, P 701-710,* 2004.

[46] Alan L. Liu, Harlan Hile, Henry A. Kautz, Gaetano Borriello, Pat A. Brown, Mark Harniss, and Kurt Johnson. Indoor wayfinding: : developing a functional interface for individuals with cognitive impairments. In *ASSETS,* pages 95-102, 2006.

[47] E. F. LoPresti, Alex Mihailidis, and N. Kirsch. Assistive technology for cognitive rehabilitation: State of the art. *Neuropsychological Rehabilitation,* 14(1/2):5-39, 2004.

[48] Rafael Munoz-Salinas, R. Medina-Carnicer, F.J. Madrid-Cuevas, and A. Carmona-Poyato. Multi-camera people tracking using evidential filters. *International Journal of Approximate Reasoning,* 50(5):732 - 749, 2009. Dempster-Shafer;Multiple-views;Particle filters;People tracking;Transferable belief model;.

[49] Min-Young Nam, Mhd Zaher Al-Sabbagh, Jung-Eun Kim, Man-Ki Yoon, Chang-Gun Lee, and Eun Yong Ha. A real-time ubiquitous system for assisted living: Combined scheduling of sensing and communication for real-time tracking. *IEEE Transactions on Computers,* 57(6):795 - 808, 2008. Prototype system;Real-time tracking;Real-time ubiquitous system;.

[50] Robert J. Orr and Gregory D. Abowd. The smart floor: a mechanism for natural user identification and tracking. In *CHI '00: CHI '00 extended abstracts on Human factors in computing systems,* pages 275-276, New York, NY, USA, 2000. ACM.

[51] Charles B. Owen, Fan Xiao, and Paul Middlin. What is the best fiducial? In *The First IEEE International Augmented Reality Toolkit Workshop,* pages 98-105, Darmstadt, Germany, September 2002.

[52] K. Pahlavan, Xinrong Li, and J. P. Makela. Indoor geolocation science and technology. *Communications Magazine, IEEE,* 40(2):112-118, 2002.

[53] Jeffrey Junfeng Pan, James T. Kwok, Qiang Yang, and Yiqiang Chen. Multidimensional vector regression for accurate and low-cost location estimation in pervasive computing. *IEEE Transactions on Knowledge and Data Engineering, v 18, n 9, September, 2006, p 1181-1193,* 2006.

[54] P. Patel and J. Holtzman. Analysis of a simple successive interference cancellation scheme ina ds/cdma system. In *Selected Areas in Communications,* volume 12, pages 796 - 807, June 1994.

[55] Nikos Petrellis, Nikos Konofaos, and George P. Alexiou. Using future position restriction rules for stabilizing the results of a noise-sensitive indoor localization system. *Optical Engineering,* 46(6), 2007. Estimation error reduction;Indoor navigation system;Infrared sensors;Noise-sensitive indoor localization system;.

[56] Nikos Petrellis, Nikos Konofaos, and George Ph. Alexiou. Target localization utilizing the success rate in infrared pattern recognition. *IEEE Sensors Journal, v 6, n 5, October, 2006, p 1355-1364,* 2006.

[57] Myllymaki Petri, Tirri Henry, Kontkanen Petri, Lahtinen Jussi, Silander Tomi, Roos Teemu, Tuominen Antti, Valtonen Kimmo, and Wettig Hannes. Location estimation in wireless telecommunication networks, 2007.

[58] V.A. Petrushin, Gang Wei, and A.V. Gershman. Multiple-camera people localization in an indoor environment. *Knowledge and Information Systems, v 10, n 2, Aug. 2006, 229-41,* 2006.

[59] Matthai Philipose, Joshua R. Smith, Bing Jiang, Alexander Mamishev, Sumit Roy, and Kishore Sundara-Rajan. Battery-free wireless identification and sensing. *IEEE Pervasive Computing, v 4, n 1, January/March, 2005, p 37-45,* 2005.

[60] Nissanka Bodhi Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *6th ACM MOBICOM,* Boston, MA, August 2000.

[61] Ilir F. Progri, William R. Michalson, and David Cyganski. An ofdm/fdma indoor geolocation system. *Navigation, Journal of the Institute of Navigation, v 51, n 2, Summer, 2004, p 133-142,* 2004.

[62] Matthew Rabinowitz and James J. Spilker Jr. Augmenting gps with television signals for reliable indoor positioning. *Navigation, Journal ofthe Institute ofNavigation, v 51, n 4, Winter, 2004, p 269-282,* 2004.

[63] Matthew Rabinowitz and James J. Spilker Jr. A new positioning system using television synchronization signals. *IEEE Transactions on Broadcasting, v 51, n 1, March, 2005, p 51-61,* 2005.

[64] Jyri Rajamaki, Petri Viinikainen, Julius Tuomisto, Thomas Sederholm, and Miika Saa-manen. Laureapop personalized indoor navigation service for specific user groups in a wlan environment. *WSEAS Transactions on Communications, v 6, n 4, April, 2007, p 524-531,* 2007.

[65] Shobha Sundar Ram, Yang Li, Adrian Lin, and Hao Ling. Doppler-based detection and tracking of humans in indoor environments. *Journal ofthe Franklin Institute,* 345(6):679 - 699, 2008.

[66] G. Retscher. Test and integration of location sensors for a multi-sensor personal navigator. *Journal ofNavigation, v 60, n 1, Jan. 2007, 107-17,* 2007.

[67] Lionel Reyero and Gilles Delisle. A pervasive indoor-outdoor positioning system. *Journal ofNetworks,* 3(8):70 - 83, 2008. GPS;GSM;Positioning;Ubiquitous;WLAN;.

[68] Andrew Rice and Robert Harle. Evaluating lateration-based positioning algorithms for fine-grained tracking. *Proceedings of the 2005 Joint Workshop on Foundations of Mobile Computing, DIALM-POMC'05, 2005, p 54-61,* 2005.

[69] T. H. Riehle, P. Lichter, and N. A. Giudice. An indoor navigation system to support the visually impaired. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE,* pages 4435-4438, Vancouver, BC, August 2008.

[70] Ranga Rodrigo, Mehrnaz Zouqi, Zhenhe Chen, and Jagath Samarabandu. Robust and efficient feature tracking for indoor navigation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,* 39(3):658 - 671, 2009. Distinctive features;Feature tracking;Motion refinement;Multihomographies;Nondistinctive features;.

[71] S. Roehr, P. Gulden, and M. Vossiek. Precise distance and velocity measurement for real time locating in multipath environments using a frequency-modulated continuous-wave secondary radar approach. *IEEE Transactions on Microwave Theory and Techniques,* 56(10):2329 - 2339, 2008. mobile unit distance measurement;mobile unit velocity measurement;real-time radio location estimation;indoor multipath propagation environment;frequency-modulated continuous-wave secondary radar approach;wireless unit;measurement system;Doppler frequency shift evaluation;radar signal;radio navigation;.

[72] F.D. Rose, D.A. Johnson, and E.A. Attree. Rehabilitation of the head-injured child: basic research and new technology. *Pediatr Rehabil,* 1(1):3-7, 1977.

[73] Hu Sheng, Lei Jian-jun, Xia Ying, Ge Jun-wei, and Bae Hae-young. Hybrid location determination technology within urban and indoor environment based on cell-id and path loss. *Journal of Chongqing University of Posts and Telecommunication (Natural Science Edition), v 16, n 5, Oct. 2004, 46-9,* 2004.

[74] R.W. Skelton, S.P. Ross, L. Nerad, and S.A. Livingstone. Human spatial navigation deficits after traumatic brain injury shown in the arena maze, a virtual morris water maze. *Brain Injury,* 20(2):189-203, 2006.

[75] MM Sohlberg, S Fickas, PF Hung, and A Fortier. A comparison of four prompt modes for route finding for community travellers with severe cognitive impairments. *Brain Injury,* 21(5):531-538, 2007.

[76] H. W. Sorenson. *Kalman filtering: Theory and applications.* IEEE press, New York, 1985.

[77] A. Stracciari, S. Lorusso, A. Delli Ponti, K. Mattarozzi, and A. Tempestini. Cognitive functions after transient topographical amnesia. *Europ J of Neurology,* 9(4):401-405, July 2002.

[78] T. Suzuki, S. Iwasaki, Y. Kobayashi, Y. Sato, and A. Sugimoto. Incorporating environment models for improving vision-based tracking of people. *Systems and Computers in Japan, v 38, n 2, Feb. 2007, 71-80,* 2007.

[79] Masahiro Takashima, Dapeng Zhao, Kentaro Yanagihara, Kiyoshi Fukui, Shigeru Fuku-naga, Shinsuke Hara, and Ken-Ichi Kitayama. Location estimation using received signal power and maximum likelihood estimation in wireless sensor networks. *Electronics and Communications in Japan, Part I: Communications (English translation of Denshi Tsushin Gakkai Ronbunshi), v 90, n 12, December, 2007, p 62-72,* 2007.

[80] S. Teller, Jiawen Chen, and H. Balakrishnan. Pervasive pose-aware applications and infrastructure. *IEEE Computer Graphics and Applications, v 23, n 4, July-Aug. 2003, 14-18,* 2003.

[81] R. Tesoriero, J.A. Gallud, M. Lozano, and V.M.R. Penichet. Using active and passive rfid technology to support indoor location-aware systems. *IEEE Transactions on Consumer Electronics,* 54(2):578 - 583, 2008. Location-aware;Location-aware systems;Mobile technologies;Passive RFID;Smart phones;System environment;.

[82] Ching-Chih Tsai, Hung-Hsing Lin, and Szu-Wei Lai. 3d posture determination of a wheeled mobile robot by fusing dead-reckoning and ultrasonic measurements. *Journal of the Chinese Institute of Engineers, Transactions of the Chinese Institute of Engi-neers,Series A/Chung-kuo Kung Ch'eng Hsuch K'an, v 28, n 6, October, 2005, Automation, p 1005-1011,* 2007.

[83] Chih-Hsiung Tseng and Sheng-Tzong Cheng. Location management scheme with wlan positioning algorithm for integrated wireless networks. *Computer Communications,* 31(18):4304 - 4311, 2008. Indoor positioning;Integrated wireless networks;Location databases;Location informations;Location management;Location management meth-ods;Location management schemes;Location-based services (LBS);Mobile clients;Mobile users;Parameter settings;Performance evaluations;Positioning algorithm;Positioning al-gorithms;Rapid growths;Received signal strength indication (RSSI);Received signal strength indications;Wireless communications;Wlan positioning;.

[84] Shinji Tsuzuki. High-speed power-line communication and its application to a localization system. *IEICE Transactions on Fundamentals ofElectronics, Communications and Computer Sciences, v E89-A, n 11, November, 2006, p 3006-3012,* 2006.

[85] J. Urena, A. Hernandez, A. Jimenez, J.M. Villadangos, M. Mazo, J.C. Garcia, J.J. Garcia, F.J. Alvarez, C. De Marziani, M.C. Perez, J.A. Jimenez, A.R. Jimenez, and F. Seco. Advanced sensorial system for an acoustic lps. *Microprocessors and Microsystems, v 31, n 6, Sep 1, 2007, Sensor Systems, p 393-401,* 2007.

[86] M. Van Asselen, R. P. Kessels, L. J. Kappelle, S. F. Neggers, C. J. Frijns, and A. Postma. Neural correlates of human wayfinding in stroke patients. *Brain Res,* 1067(1):229-238, January 2006.

[87] N. Vaughan, D. Agner, and D.M. Clinchot. Perseveration and wandering as a predictor variable after brain injury. *Brain Injury,* 11(11):815-819, 1997.

[88] S. Venkatesh and R.M. Buehrer. Non-line-of-sight identification in ultra-wideband systems based on received signal statistics. *IET Microwaves, Antennas and Propagation,* 1(6):1120- 1130, 2007. Non-line-of-sight (NLOS);Statistical information;Ultra-wideband (UWB);.

[89] Xue Wang and Sheng Wang. Collaborative signal processing for target tracking in distributed wireless sensor networks. *Journal ofParallel and Distributed Computing, v 67, n 5, May, 2007, p 501-515,* 2007.

[90] Roy Want, Andy Hopper, Veronica Falcao, and Jon Gibbons. The active badge location system. Technical Report 92.1, ORL, 24a Trumpington Street, Cambridge CB2 1QA, 1992.

[91] R. Watson, G. Lachapelle, R. Klukas, S. Turunen, S. Pietila, and I. Halivaara. Investigating gps signals indoors with extreme high-sensitivity detection techniques. *Navigation. Journal ofthe Institute ofNavigation, v 52, n 4, Winter 2005-2006, 199-213,* 2006.

[92] Zhe Xiang, Song Song, Jin Chen, Hao Wang, Jian Huang, and Xingxin Gao. A wireless lan-based indoor positioning technology. *IBM Journal ofResearch and Development, v 48, n 5-6, September/November, 2004, p 617-626,* 2004.

[93] De Xu and Min Tan. Accurate positioning in real time for mobile robot. *Zidonghua Xuebao/Acta Automatica Sinica, v 29, n 5, September, 2003, p 716-725,* 2003.

[94] De Xu, Zhi-Guo Tu, and Min Tan. Study on visual positioning based on homography for indoor mobile robot. *Zidonghua Xuebao/Acta Automatica Sinica, v 31, n 3, May, 2005, p 464-469,* 2005.

[95] Qiang Yang, Sinno Jialin Pan, Vincent Wenchen Zheng, Hisashi Kashima, Shoko Suzuki, Shohei Hido, Yuta Tsuboi, Toshihiro Takahashi, Tsuyoshi Id, Rikiya Takahashi, Akira Tajima, Yuichi Katori, Yang Qu, Chun Li, Xi-Zhao Wang, Feng Guo, Xianghui Gao, Zhuo Sun, Juan Qi, Junfa Liu, and Yiqiang Chen. Estimating location using wi-fi. *IEEE Intelligent Systems,* 23(1):8 - 9, 2008. Indoor location;Radio signal strength;Transfer learning;.

[96] Jaegeol Yim, Chansik Park, Jaehun Joo, and Seunghwan Jeong. Extended kalman filter for wireless lan based indoor positioning. *Decision Support Systems,* 45(4):960 - 971, 2008. Bayesian method;Decision tree;Extended Kalman Filter;Fingerprinting method;Indoor positioning;K-NN;Trilateration;Wireless Local Area Network;.

[97] Wojciech Zajdel and Ben J.A. Krose. A sequential bayesian algorithm for surveillance with nonoverlapping cameras. *International Journal of Pattern Recognition and Artificial Intelligence, v 19, n 8, December, 2005, p 977-996,* 2005.

[98] G.V. Zaruba, M. Huber, F.A. Kamangar, and I. Chlamtac. Indoor location tracking using rssi readings from a single wi-fi access point. *Wireless Networks, v 13, n 2, April, 2007, p 221-235,* 2007.

[99] Z. Zhang, C.L. Law, and E. Gunawan. Multipath mitigation technique based on partial autocorrelation function. *Wireless Personal Communications, v 41, n 1, April, 2007, p 145-154,* 2007.

[100] YunFei Zu, MunGyu Choi, and JangMyung Lee. An indoor localization system in a multiblock workspace. *Artificial Life and Robotics,* 13(2):493 - 499, 2009. Autonomous navigations;Fault detection and isolations;Indoor localiza-tion;Localization schemes;Mobile objects;Multiblock;Position estimations;Precision anal-ysis;Service robots;Ubiquitous technologies;Ultrasonic attenuations;.

4

Framework for Visual Analysis of User Behaviour in Ambient Intelligence Environment

[1]Ivo Maly, [2]Jan Curin, [1]Pavel Slavik and [2]Jan Kleindienst
[1]*Czech Technical University in Prague, Faculty of Electrical Engineering, Czech Republic;*
[2]*IBM Czech Republic, Voice Technologies and Systems*

1. Introduction

Ambient intelligence is a rapidly developing field that is led by the necessity of ubiquitous computing. People are surrounded by computers in most of their daily activities; i.e. in cars, homes, offices and public areas, and supplied by various services on mobile phones, PDAs, or interactive kiosks. There are several projects that demonstrate potentials of ubiquitous computing, e.g. in a museum (Gallud et al., 2007), in a home (i2home, 2009), or in an office (Waibel, 2009).

In our case, the ambient intelligence environment consists of two components, a mechanism for detection of user behaviour and an application that supports specific user scenario or set of scenarios (ambient intelligence applications). Along with increased complexity of the systems and user workflows in an ambient intelligence environment, the usability and functionality of both components must be ensured. Traditional approaches of usability testing are suitable for testing individual components in an isolated manner. However, usability of the whole system can significantly differ from the usability of individual components. A reason for testing of individual components is the inability to setup a complete environment of supporting technologies as they are still under development or yet non-existent in early phases of system design. Unfortunately, these phases are the ones where a correct design decisions are crucial, because they influence further development of the whole system.

In this paper we introduce a framework for testing applications in ambient intelligence environments. The concept is derived from the methodologies of desktop application usability testing combined with evaluation of user behaviour in an ambient intelligence environment. It is based on our experience with user behaviour recording, which is visualized and investigated mainly in a virtual environment. The visualization serves as a tool for analyzing complex situations and detection of usability issues in ambient intelligence applications.

2. Motivation

Usability testing methodologies for desktop applications are quite well known and well described e.g. by (Rubin, 1994) or (Nielsen, 1994). The methodologies typically use observation of the test participant, while solving a task from the task list where the tasks use the tested application. Indirect observation is recorded and can be replayed several times. Typically a small amount of participants (4-12) are tested and mainly qualitative issues are evaluated, interpreted, and used for suggested improvements. There may also be other data sources involved in application usability testing; typically application log files (Ivory & Hearst, 2001), where information about user interaction with application is stored. Information from such sources can help to thoroughly analyze particular parts of the application or the test.

Many modifications of this methodology have appeared. One modification that is significant for our ambient intelligence environment testing methodology is remote usability testing (Paterno et al., 2007). This methodology usually does not use observation of participant interaction but rather derive usability issues from user interaction data generated by application. This approach is suitable when observation is not possible due to technical limitations, e.g. test participant and usability expert are at different places. An advantage of the desktop application in this methodology is that we can quite accurately estimate context of use, based on target user group knowledge. This allows us also to focus more on the observation of user interactions, supplemented with user thoughts acquired by the thinking aloud technique (Rubin, 1994).

On the other hand, applications for ambient intelligence environments are much more context sensitive. Context can change several times during a single test and therefore we have to give the evaluator all the important context information. Observation of the user in such an environment is also problematic. Analysis from the recording from observation of participant interaction can be complicated when more cameras are used in a complex test setup. Moreover analysis of video recording is very time consuming. These issues can be solved by transformation of the participant behaviour and interaction into the visual representation (e.g. virtual environment) that is used for further test analysis and application usability study. There are also several other motivations that support the needs to be solved in order to improve process of user behaviour analysis.

The first motivation is the ability to analyze huge amounts of data that is gathered from the tested application and from the supporting infrastructure of the ambient environment. It is very complicated to analyze the data in its raw format such as a text file. The visualization in the virtual environment can be one of the suitable data transformations.

The second motivation is unification of different test data for direct comparison. Traditionally, we can face to the problem of comparison of data from ambient environment sensors (e.g. time snapshots of the ambient intelligence environment state), video recording from environments where no other sensors are available and user scenarios created artificially (e.g. text description). Therefore we should be able to transform all those data types to one virtual environment.

The third motivation is connected with ethical issues. The process of usability testing obeys strong ethical codices, which also keep to the laws that are part of the legal system.

Typically, it is not allowed to publish a video of the user from the test, where he/she can be recognized, without his/her permission. In desktop application usability testing the problem can be easily solved. The video from a desktop application focuses on the user interaction with the application, which means that the application is the object being recorded and the user usually does not appear in the video (the user can be recorded using another camera). In the case of mobile application usability testing, the user is part of the environment and his/her interaction and behaviour is being investigated. Such a video recording is then not usable as material for application usability issue presentation. Instead of making the user anonymous in the video recording the user can be made anonymous by the process of data transformation into the virtual environment where the user is represented in the form of avatar.

3. Related Projects

There are several projects that also deal with evaluation of ambient or ubiquitous applications. Project UbiWise (Barton & Vijayaraghavan, 2002) tries to simulate ubiquitous environment equipped with devices that communicate with each other. It focuses on simulation of hardware and low level software of the ubiquitous environment. Another project that has similar motivation as UbiWise is project UbiREAL (Nishikawa et al., 2006). In this article the authors present a tool for testing the correctness of virtual or real UPnP (Universal Plug and Play) protocol based appliances connected into a network. The authors focus on the presence of the appropriate appliance, controller functionality and logic instead of the user behaviour. Project 3DSim (Nazari Shirehjini & Klar, 2005) is tool for rapid prototyping of ambient intelligence applications. This tool uses also UPnP protocol (Nazari Shirehjini, 2005) for inter device communication. It is similar to both UbiWise and UbiREAL projects and it has some advantages like photo-realistic impression of visualization and support simple visualisation of human interaction in the environment based on recognized data. Project P-VoT (Seo et al., 2005) is component based interactive toolkit for virtual environments to quickly build a virtual ubiquitous environment. This tool allows rapid creation of the environment with possibilities of detailed device logic specifications.

We have identified two reliable pervasive computing systems which may serve as context acquisition and modelling platforms for multimodal services: already mentioned UbiREAL and functionally similar Context Toolkit (Dey et al., 2001). But in contrast to these systems, which integrate contextual information directly from various sensors, we needed a system which relies on information provided by more complex perceptual components, i.e. more complex context-acquisition components such as person trackers and speech recognizers. A body tracker might be at once capable of detecting location, heading and posture of persons, identifying them, and tracking subjects of their interest. Such scenarios led us to separate the perceptual components layer from the layer that deals with higher abstraction – the situation modelling. Defining and modelling situations based on a wide range of context-acquisition components was not supported by other environments such as UbiREAL and Context Toolkit, so we decided to implement a new framework.

4. Overview of the Framework

4.1 Introduction

Evaluation of applications in ambient environment is a complex issue. Therefore in next sections we will introduce framework that was developed to encapsulate all tasks necessary for analysis of user behaviour as part of usability evaluation of applications in ambient intelligence environment. Scheme of the framework is in the Fig. 1. Data from detection mechanism, that consists of sets of sensors (e.g. noise level detectors) placed in the ambient intelligence environment, are imported to the framework's internal data storage either directly or through the Testing of detection mechanism accuracy task. The data can be also created and edited in User scenario creation and editing task. These two tasks are depicted by rectangle with rounded corners and represent group of tasks that edit raw data. Once the data are in internal format they are persisted or immediately used for visualization in next two tasks. These two tasks are Evaluation of single user scenarios and Evaluation of cooperative scenarios. They represent high level task that analyze the data as user behaviour in the ambient intelligence environment. In next chapters we will describe each task in detail.

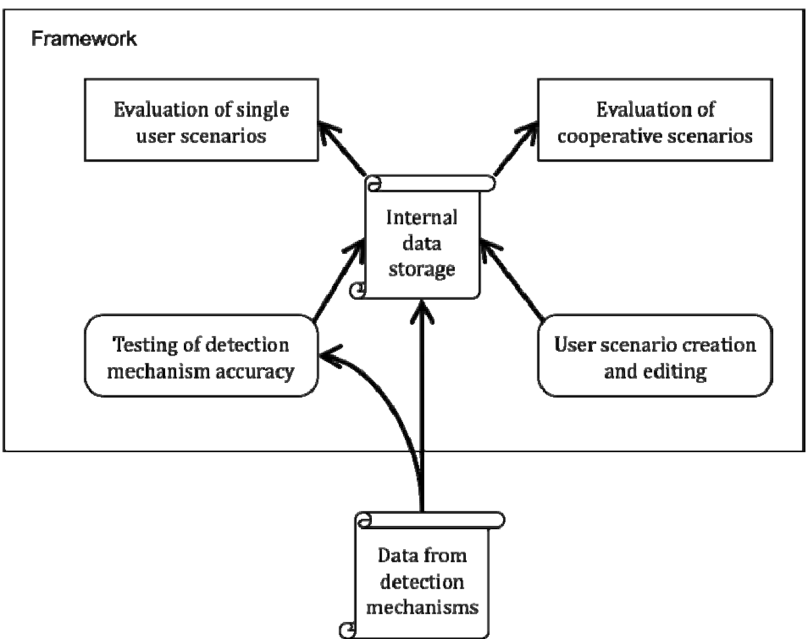

Fig. 1. Scheme of the framework.

4.2 Testing of Detection Mechanism Accuracy

Testing of detection mechanism accuracy is the lowest type of test. Aim of this test is analysis of accuracy of particular sensors and algorithms that process data from those sensors. Ambient intelligence applications strongly rely on those detection mechanisms and their algorithms and incorrect output of the detection process may highly influence behaviour of the system and usability of the application for the user.

Typical use case is evaluation of data from one or several detection mechanisms and comparison of them with either video recording or with manually revised data usually extracted from video recording. Output of this task is either report about detection mechanism accuracy or correction of data produced by the detection that are then used for evaluation of user behaviour or application usability in next types of evaluation.

4.3 User Scenario Creation and Editing

At the early stage of testing the sensors are not always available. Therefore it is not possible to collect data from them and analyse user behaviour in the environment. Therefore the framework must contain task where the data are created and edited.

Typical use case is creation of single user scenario from external formats (e.g. audio/video recording), creation of user data by expert creation of cooperative scenario from external formats (e.g. audio/video recording) and enhancement of the data from detection mechanism with additional information. Such information may be for example interaction with objects that is not detected by any detection mechanism. Execution of this task is demonstrated in section 7.1. Output of this task is set of scenarios.

4.4 Evaluation of Single User Scenarios

This type of evaluation is the simplest case of usability evaluation of the application. We investigate behaviour of the user according to the predefined scenario. We should compare the user behaviour with pre-recorded data created by expert, in order to find deviations from the expected behaviour. Expert data may be gathered directly from the ambient intelligence environment and eventually polished in task testing of detection mechanism accuracy. Alternatively they can be created by the framework in task user scenario creation and editing, where the simulation of the user interaction in the ambient intelligence environment is performed by expert.

There are several possible use cases how to perform this task. The first use case is simple evaluation of one user. The second type of use case is comparison of one user with the data created by expert. The third use case is comparison of several users between each other. Finally fourth use case is comparison of several users and data from expert between each other. Execution of this task is demonstrated in section 7.3.

4.5 Evaluation of Cooperative Scenarios

Cooperative scenarios are the most complicated scenarios for evaluation in all environments. Goal of this test is to evaluate whether the application correctly supports cooperation of individual users in their scenario. Currently the only use case supported is evaluation of one cooperative session. Execution of this task is demonstrated in section 7.2.

5. Visualization of User Behaviour

5.1 Introduction

When performing various user tests and other experiments that deal with user behaviour we always obtain large data sets. These data should be evaluated and some conclusions should be drawn in order to improve the design of particular user interface. Due to the large amount of data it is mostly impossible to perform analysis of these data without deeper insight (to understand mutual relations between various variables or between data subsets in the given data set). Solution to this problem is to use an appropriate visualization technique by means of which we can get a deep insight into data investigated (and thus to make some relevant conclusions).

There exists a large amount of various visualization techniques by means of which it is possible to visualize data (or information of various kind) – these techniques are of interest in a special branch of computer graphics, particularly, information visualization (Spence, 2000). In our research we had to select and modify some approaches in order to be appropriate for our purpose that is visualization of user behaviour in ambient intelligence environment.

5.2 Survey of Visualization Techniques

At first we have to define what kind of activity we are going to investigate and visualize. As our research has been concerned with user interaction in ambient intelligence environment we had to investigate (and visualize) three main aspects of user behaviour:

- visualization of user interaction with devices (log file with user activities when solving a particular task).
- visualization of the user movement in the scene (visualization of the user movement in the scene gives us a global picture about the activities of the user)
- visualization of the user movement and behaviour (such a visualization offers a complex picture about the user when solving particular task in ambient environment)

In Fig. 2 we can see that these three aspects of user behaviour form hierarchy – here we can speak about a sort of level of detail in which we would like to investigate the data obtained. In (Mikovec et al., 2007) we showed that this concept corresponds with three types of visualization. In our framework we also choose appropriate visualization method when investigating particular user behaviour. In such a way we can analyze both individual approach to the solution of task (that consists of several subtasks) and movement of the user in the scene where the task should be solved. In further text these three visualization methods, that are suitable for visualization of three data types given above, will be presented.

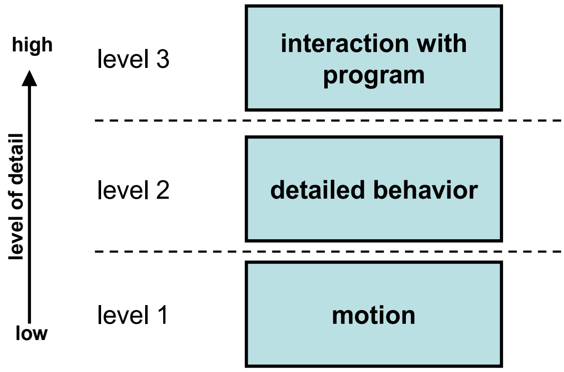

Fig. 2. Level of detail of user behaviour visualization.

In all visualization methods (levels) we tried to analyze and describe the user behaviour based on data set, which was acquired through observation of user interaction with application (stored as interaction log file). Data set is created by means of collected data that reflect the user activity when the user performs some typical tasks. These activities (performing certain task or collection of tasks) are performed in the framework of usability testing (Rubin, 1994) (the user interface designed is tested by several of users who perform some typical tasks). The data about the performance of users are collected using various methods – mostly observation of the user or recording the user activities by means of various tools (recording the individual interaction steps during particular task solving etc.). In order to understand such a complex data we need to have a proper visualization tool that is interactive and allows us to select proper view on the data.

For the Level 3 we can use visualization in form of timelines described in (Harrison et al. 1994) and (Malý & Slavík, 2007). These visualizations allow visualizing data both from standard usability testing and also from ambient intelligence environment. The timeline (Fig. 3) is presented as a sequence of annotations ordered according to the start time of the annotation. Each annotation represents one activity of the user during the usability testing. Annotations are grouped into categories based on the severity of the annotated problem. The severity is represented by different colour of the annotation – red is more complicated for the user, white is easier for the user). The visualization also shows annotations together with task tree (Paterno et al., 1997) (that describes in a formal way the structure of the task performed) in one view. The task tree in visualization shows by green rectangles at which time certain activity (e.g. subtask) begun and when this activity was finished.

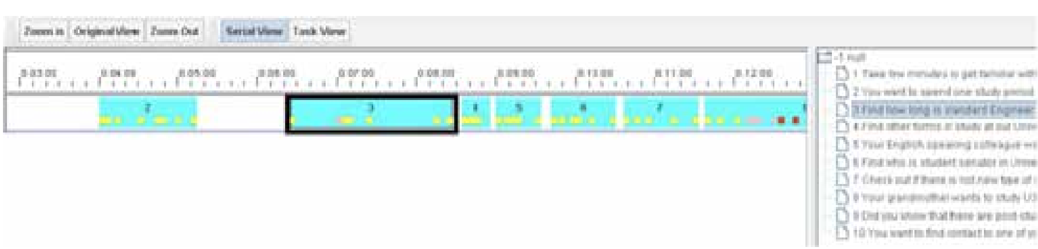

Fig. 3. Timeline visualization of user interaction. Colour of annotation represents severity of the problem.

As we are in our research concentrated on a specific environment – ambient environment – we should have some idea about the user motion in this environment. For the sake of simplicity we can imagine that a user walks in an exhibition hall, in a gallery, in a museum etc. When following her/his path we can discover some patterns (like concentration on specific areas – where e.g., are exhibited topics of her/his interest like some products, pictures etc.). In such a way it will be possible to discover attitudes of individual users (this may help to define groups of users that share common interests etc.). This kind of evaluation of the user behaviour, representing Level 1 in Fig. 2, is based on visual analysis of the acquired data. The user is trying to get from the position A into the position B. What is interesting is the investigation of strategy for selection of the trajectory that has been chosen by the user. This information can give important information about the user background on which the selection of strategy was based on. The visual analysis, see Fig. 4, results in determination of spots in ambient environment where the user spent more time or where the user appears frequently. By means of proper visualization tool it is possible to identify these spots in a very short time.

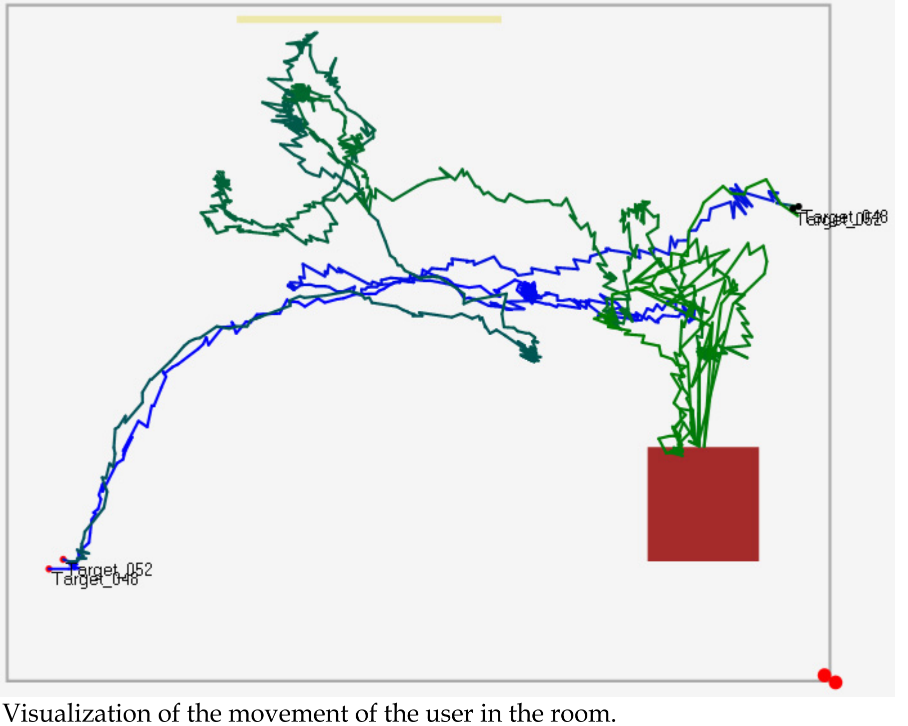

Fig. 4. Visualization of the movement of the user in the room.

Fig. 5. 3D visualization of the user interaction in the room.

The fact that the third dimension gives us many advantages when visualizing data about the user behaviour resulted in development of a new visualization technique, which represents Level 2 in Fig. 2. This third visualization technique allows us to visualize user movement in an ambient environment – e.g. in a smart room. The data for visualization are from real sensors or they are manually transformed from video recordings into simulated user scenario that is interpreted in a simulator. The output of the simulator is visualization of the user behaviour in 3D. Such a visualization is performed in virtual environment where we can manipulate both with the virtual environment (zooming, rotations with camera, …) and with data visualized (data about motion or behaviour of the user). Thanks to that we can get much better idea about user's intentions, habits, behavioural patterns, etc. Such an approach provides higher (semantic) abstraction in the situation modelling module, and presents it to the user graphically in a 3D scene, see the Fig. 5.

When summing up the properties of the visualizations described it is possible to say that the visualizations presented create a scalable system that allows user to visualize complex data obtained in the framework of usability testing. The possibility to select proper visualization makes the evaluation of data collected more flexible and more efficient (it is possible to evaluate data on the proper level of detail).

6. Applications Used in the Framework

6.1 Introduction

Our framework is based on our experience with development and usage of two applications for analysis of user behaviour in ambient intelligence environment. The first application is

SitCom - Situation composer. The second tool is USEd - User Scenario Editor. Description of both tools is in next sections, followed by experiences with these tools in several use cases.

6.2. SitCom – Situation Composer

SitCom (Fleury et al., 2007) serves as a simulation runtime. i.e. the core tool supporting the whole development cycle of multimodal perceptual systems.

6.2.1 Smart Environment Architecture

Before we introduce the SitCom tool, we should describe basic terms and abstraction architecture of smart environments. Sensors deliver their information in a streaming fashion, as do video camera for video data or noise level detectors for a given value of the noise level. The perceptual components, which do some combination of the sensor data to render it at a higher semantic level, still propagate this information, after processing, in a streaming fashion. A body tracker will update the number of detected bodies, and their locations. A room noise classifier will deliver the new label representing a person in the room as soon as it has processed the information from sensors. In that sense, the data is going through the layers as in a pipeline. As depicted in Fig. 6, the information flows through the following layers: sensors, perceptual components, situation modelling, and services.

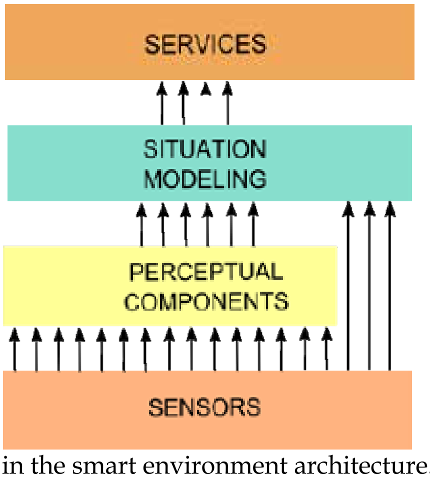

Fig. 6. Levels of abstraction in the smart environment architecture.

Perceptual components provide information about *entities*, which are mapped one-to-one to real world objects. These entities have a set of attributes, which may change over time. These attributes are modelled as *property streams*, with an update frequency, which depends on the type of attribute. For example, a person's ID will most likely change only a couple of times, depending on the accuracy of the person identifier technology, whereas the person's location will most likely be updated often.

The *situation modelling* layer is where the transformation of a set of facts about the environment into a set of situations is taking place (Crowley et al., 2002). The situation model will provide information about *situations*, which consist of a set of *states* and their associated *transitions*. The detection of the current state and the detection of state changes is implemented using several techniques, and is extendable. Currently, we have a rule-based engine, several statistical methods, and a very simple hardwired logic. Each situation is embodied in a *situation machine*, which is a Java module in SitCom, and only the external hull of the situation machine is visible. This uniformity of public interface for situation machines is a key aspect for the flexibility of the simulation environment. The actual set of used situation machines is application dependent.

The service layer contains smart environment's applications provided to the users or participants, examples of such applications are given in use cases in sections 7.2 and 7.3. More details about this architecture used herein can be found in (Dimakis et al., 2008).

6.3 The SitCom Tool

SitCom (Situation Composer for Smart Environments) is a 3D simulator tool and runtime for development of context-aware applications and services. Context-aware applications draw data from the surrounding environment (such as there is an ongoing meeting in the room, a person location, body posture, etc.) and their behaviour depends on the respective situation (e.g. while in meeting silent the phone). In SitCom, the environment characteristics are captured by Situations Models that receive input events from a) real sensor inputs (cameras, microphones, proximity sensors...), b) simulated data, and c) combination of real and simulated input. SitCom allows combining the situation models into hierarchies to provide event filtering, aggregation, induction and thus construct higher-level meaning and reasoning functionality. Through its IDE, SitCom allows to load, run or modify various scenarios with their subsequent realistic rendering in 3D virtual environment. These scenarios can be re-played to invoke situations relevant for the application behaviour, and thus provide mechanism for systematic testing of the context-aware application under different environmental conditions. The key feature is that we keep the portability between virtual and real devices.

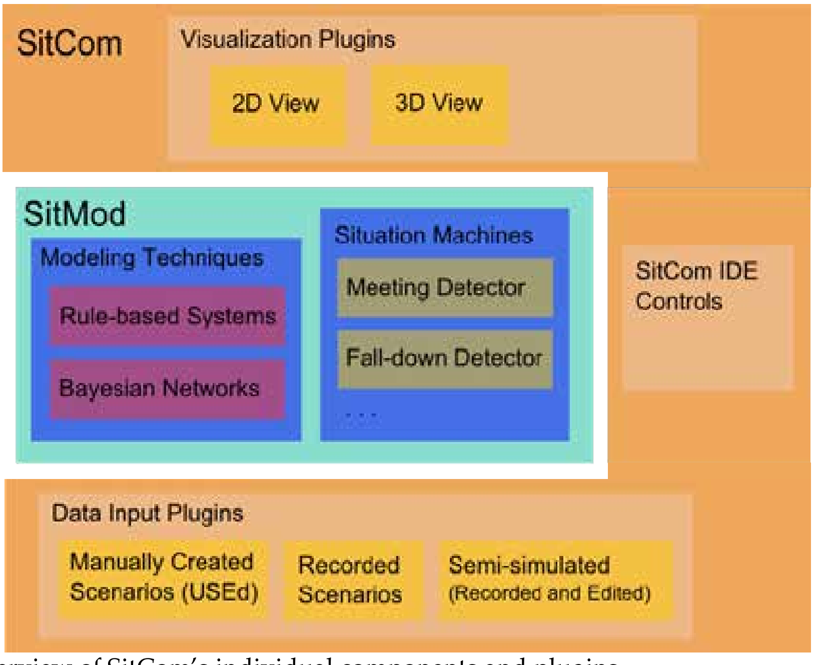

Fig. 7. Overview of SitCom's individual components and plugins.

6.3.1 SitCom's Components and Plugins

In the reference architecture model, SitCom itself plugs into the layer of situation modelling. So it sees the service layer as its upper layer, and the perceptual components as its lower layer. Also, as SitCom itself is a framework for modelling systems enabling plugging diverse situation modelling technologies, it is split into two distinct parts: the simulation framework, named SitCom, and the actual situation modelling being developed, named SitMod (Situation Modelling). In Fig. 7, the two parts are clearly separated. SitMod (and modules plugged into it) is the portion of the system to be deployed into the field. During simulation, but also once deployed, the SitMod run-time runs situation models for multiple and possibly unrelated scenarios. We found it very useful to be able to switch back and forth between simulated (manually created) and recorded data, as for debugging some situations, we can run the experiment up to a point, and then we can interfere with the recorded data using synthetic data, so as to test the situation model and the service. This makes the test of "what if" scenarios very quick and simple.

In the Fig. 8 there is main window of SitCom application. We can see visualization of the room in 3D virtual environment (Room-3D visualization plug-in) and in 2D visualization (Room-2D visualization plug-in). The visualization is generated based on data from input plug-ins. In the lower left part, there are controls for replay of the simulation and controls for handling of input plug-ins. On the right, there is video recording of the room, synchronised with both visualizations.

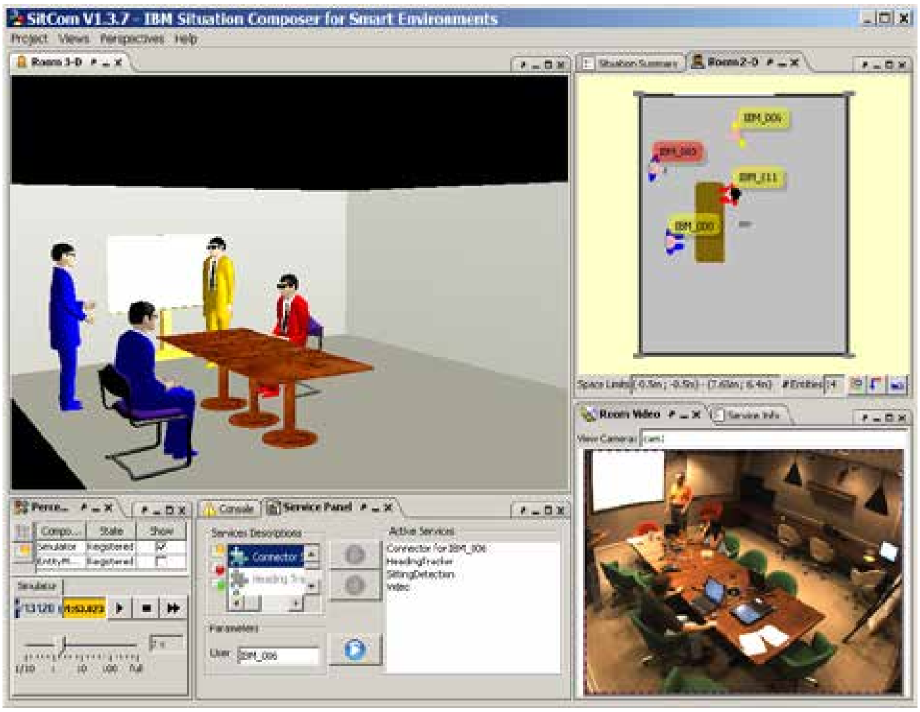

Fig. 8. Meeting scenario visualization using the SitCom tool.

6.4 USEd – User Scenario Editor

In contrast to SitCom, USEd is used to create, edit and tune user scenarios in ambient intelligence environment (especially in smart environments).

6.4.1 Main Principles

For the purpose of user scenario creation and editing task we developed the USEd, the **U**ser **S**cenario **Ed**itor. This tool supports most of the use cases presented in the task user scenario creation and editing, so it is possible to create single user scenario from external formats (e.g. audio/video recording), create single user scenario by usability expert, create cooperative scenario from external formats (e.g. audio/video recording).

The User Scenario Editor is an application for user scenario creation. A user scenario represents behaviour of the user in the ambient intelligence environment during usability evaluation. This editor is data compatible with SitCom, because it uses a SitCom environment description file for virtual environment description and one of the outputs of the editor is a SitCom person description file. Therefore scenarios recorded in the USEd can be investigated in SitCom runtime. Advantage of USEd is that it stores the user scenario as events instead of time snapshots of the environment status as SitCom. Therefore it is much easier to create and edit the scenario in USEd then in SitCom. USEd supports also simple verification of the created scenario so we can prepare the scenario without necessity to use SitCom to evaluate the scenario.

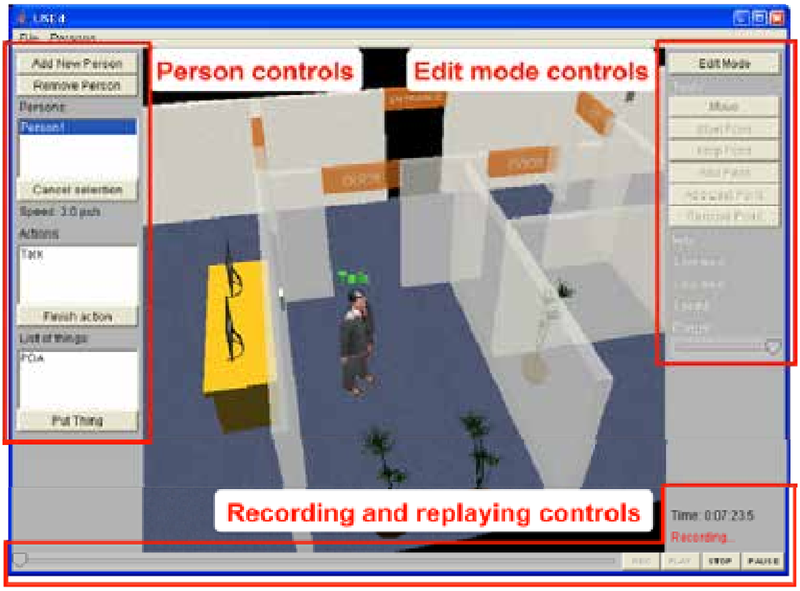

Fig. 9. Main window of USEd editor with.

The main screen of the USEd is divided into four parts, see Fig. 9. In the middle, the interactive virtual environment is displayed. USEd allows zooming, panning and rotation of the environment to allow full control during scenario editing. At the bottom, there are buttons for starting and stopping of the recording and for starting and stopping replaying of the recorded scenario. There is also the timeline showing time of the recorded scenario and allowing time shift during replaying. On the left, there are person controls for online recording. We can add or remove a person from the recorded scenario and we can select which person is controlled in case there is more than one person in the scene. There is also a list of actions the person is currently performing and list of things the person is currently holding. Two buttons are used for finishing the selected action or for putting the selected thing. On the right, there are edit mode controls with button toggling between data recording and editing.

6.4.2 Scenarios Creation and Editing

Scenario creation is the activity when a new scenario is created. It is started by pressing the REC button in bottom right corner. After that, timing begins and we are recording a new scenario. We add a new person by pressing the Add New Person button. A dialog for setting the name of the person appears and after entering the name the person appears in front of the starting point Entrance. Person movement is controlled by the clicking into the scene. After the clicking the person tries to move to the position where we clicked. Detection collision and path finding algorithms were implemented to simplify the scenario creation. Interaction of the person with other objects in the scene is performed by right mouse button clicking on the object. After that, a dialog menu with possible actions appears and we can select the desired action. There are two types of action; immediate actions and long lasting actions. The difference is that long lasting actions appear in the action list of the person and must be explicitly finished compared to immediate actions that are executed immediately and do not appear in the action list of the person.

Editing is an activity suitable for correction of the recorded scenario. It is activated by pressing the Edit Mode button in the upper right corner of the application. In Edit mode the virtual environment changes (see Fig. 10), the person disappears and an individual person path is displayed instead. Also new buttons that switch between different user interaction modes are available. Edit mode supports movement of the navigation points, adding and removing points in the path and also adding new points to the path. Timeline controls the segment of the path that can be edited. This is extremely important in case the path is very dense in a particular area and helps us not to operate with wrong points of the path.

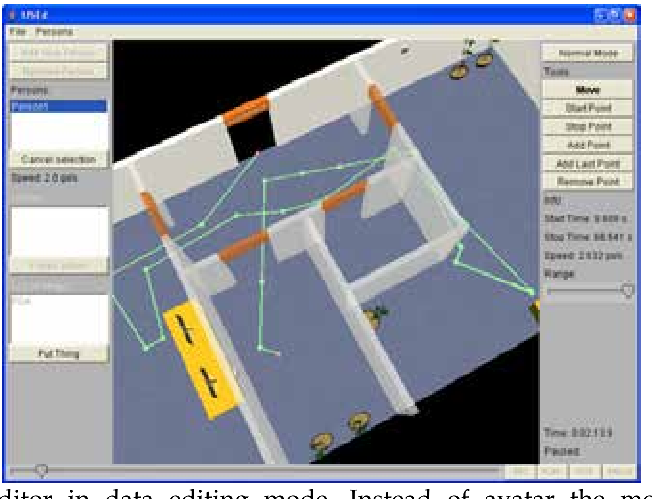

Fig. 10. USEd editor in data editing mode. Instead of avatar the movement path is displayed.

Collaborative scenario creation can be achieved by two different approaches. The first approach is one-by-one scenario recording, where one person is being recorded while other persons that were recorded are replayed to help with synchronization. The second approach is multiple persons recording, when all persons are recorded simultaneously. Using edit mode the actions are shifted to appear concurrently.

7. Use Cases

7.1 Inventory Use Case

The Inventory use case was used to evaluate the ability of the USEd editor to transform data from video recording into the virtual environment and create expert evaluation. Then, these two types of transformations were compared both from ease of creation and a quality of results point of view.

This use case was testing two parts of the framework, the task user scenario creation and editing and internal data storage.

7.1.1 Use Case Description

In this scenario the user is doing inventory in the smart environment with a PDA. The PDA has a barcode reader that allows for fast recognition of inventoried items. If the item is not in the correct place or in the correct room, the user can move the object in the application to the correct place. The scenario consists of the following steps:

- User enters the room.
- User looks around.
- User goes to the set of three LCD displays.
- User checks the displays and makes appropriate changes in the application using his/her PDA.
- User leaves the room.

7.1.2 Data Acquisition Setup

Expert evaluation data were created before the video of the real user was created. It is based on the scenario steps with estimated interactions and timings of user actions. The expert had no limit for data creation to avoid any influence.

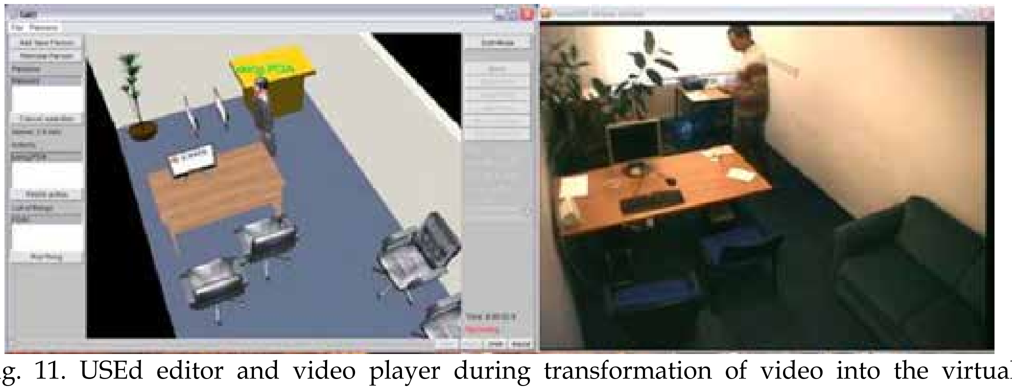

Fig. 11. USEd editor and video player during transformation of video into the virtual environment.

One video of the use case was recorded. Then 5 people tried to transform the video into the virtual environment using USEd. The computer setup is shown in Fig. 11. The USEd editor is on the left; the video player with a recorded scenario is on the right. Each person had to record the video in real time with an additional 5 minutes for corrections.

7.1.3 Use Case Results

All the people were able to transform the video into the virtual environment. There were some small usability issues of the USEd editor that will be corrected during future development. All the videos were similar, only a limited time drift of action start time appeared. Also, the algorithm for collision detection and path finding helped to unify the movement of the avatars.

7.2 Meeting in Smart Room Use Case

The meeting use case arose from actual needs in a project called CHIL (Computers in the Human Interaction Loop) (Waibel, 2009). Objectives of the CHIL project were to create a smart environment (in this case room) in which computers serve humans that focus on interacting with other humans. The use of expert-created scenarios at the beginning of the project allowed us to start development of services prior to setting-up all the required technologies. Later when the technologies were ready, it gives us the possibility to compare behaviour of the same service using either expert-created or real data. From the framework point of view, we were evaluating cooperative scenario, using real data and simulated data.

In our use-case example, we consider a connector scenario proposed and exploited in (Danninger et al., 2005). The connector service is responsible for detecting acceptable interruptions (phone call, SMS, targeted audio …) of a particular person in the smart room. During the meeting, for example, a member of the audience might be interrupted by a message during the presentation, whereas the service blocks any calls for the meeting presenter.

7.2.1 Use Case Description

The meeting state detector is implemented as a situation model in the SitCom framework (described in section 6.2). The situation modelling layer is the place where the situation context received from audio and video sensors is processed and modelled. The context information acquired by the components at this layer helps services to respond better to varying user activities and environmental changes. For example, the situation modelling answers questions such as: Is there a meeting going on in the smart room? Who is the person speaking at the whiteboard? Has this person been in the room before? Schemas in Fig. 12 correspond to the setup for the connector service scenario. In this case, the situation model provides the information about the current state of the meeting to the service.

7.2.2 Bootstrapping Situation Models by Scenarios Created in USEd

When a project is currently in its first stage, the "emerging" technologies planned to be used are usually under development, but service providers are already eager to try prototypes and discuss deployment of feature functions with possible clients. In such cases, the possibility to create a virtual space capable of simulating the technologies helps and speeds-up the whole development cycle of the project.

Schema in Fig. 12 shows the situation model created for a meeting state recognition task. We have investigated numerous experiments in order to find a suitable statistical method. According to the needs of CHIL services, the methods were applied to a task of detecting the state of the meeting using labels provided by the perceptual components. We use the facts produced by perceptual components (about people's presence, location, pose, and voice activity in the room) to determine the meeting states. These facts may themselves be the result of the analysis of multiple sensors.

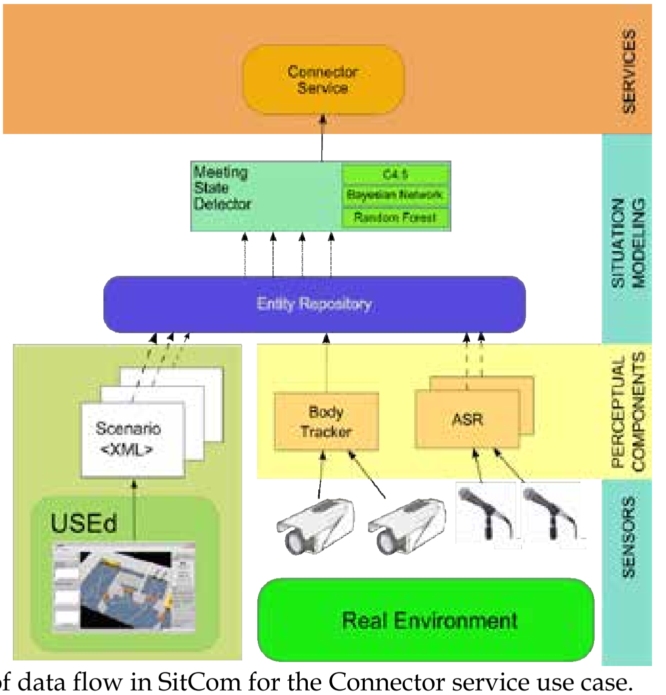

Fig. 12. Schema of data flow in SitCom for the Connector service use case.

7.2.3 Use Case Results

The expert-created scenarios (generated by the precedent of USEd) were used to bootstrap the initial work and to obtain first results. Identified methods and feature sets were then later used with seminar data, particularly the CHIL seminar 2007 corpus for both model training and evaluation. Fig. 8 shows visualization of recorded seminar data in the SitCom tool. A positive experience was that the feature sets and their parameters built and tuned on expert-created scenarios needed only minor changes when applied to real data. The differences were in selected time thresholds and intervals; the expert-created scenarios were not as dynamic as the real recording. Yet another minor difference was in expected and real speech activity among meeting participants; the expert-created scenarios contained many more interruptions and noisy conversations than the real case. We have used five manually crated scenarios of total length 1 hour 25 minutes; the real recordings comprised 6 scenarios of 1 hour 50 minutes length.

7.3 Home for Elderly Use Case

This use case is the most complex evaluation of our framework. First of all, we are testing detection mechanism accuracy and then we are evaluating single user scenario using both real and simulated data.

7.3.1 Use Case Description

Another more recent use case based on the approach advocated in this paper is Netcarity project. Netcarity is an ongoing integrated project supported by the European Community under the Sixth Framework Programme (IST-2006-045508). The aim of this project is investigation and testing of technologies which will help elderly people to improve their well being, independence, safety and health at home.

Fig. 13. SitCom's 3D visualization of a smart home.

The idea of this project is to equip homes of Netcarity clients by a variable set of perceptual technologies and connect them to the central point providing various types of services. The most important activity of the central service point, called Netcarity Server (NCS), is to "keep an eye" on the client health and safety exploiting different perceptual technologies, i.e. acoustic event detector, body tracker.

This project is currently in its first stage when the technologies are still under development, but service providers are already eager to try prototypes and discuss deployment of feature functions with possible clients. We have created a SitCom virtual smart home and several scenarios simulating essential situations in inhabitants' life, see Fig. 13. The following sensors, actuators and perceptual components are being simulated in the virtual home environment:

- Motion Detectors are installed in each room, monitoring movement in the space.
- Fire Detector sensor detecting fire in the flat.
- Gas Detector sensor detecting leaking gas in the flat.
- Remote Door Lock sensor/actuator capable of reporting open/closed status of the door and locking/unlocking the door's lock.
- Scene Acoustic Analyzer analyzing noise and sounds in the scene, such as, someone is walking, speaking, laughing, door is opening/closing.
- Body Tracker is a video-based tracker providing 2D coordinates for each person in the room.

- Talking Head actuator giving audio/visual feedback to the inhabitant. It serves as a communication channel from the Netcarity server and as a humanoid interface to other services.

The example of situation machine is:

- Fall-down Detector perceptual component detecting that the inhabitant may be in dangerous situation, for example he/she felt down and is unable to call for help. It uses and combines information from different sensors and detectors.

7.3.2 Bootstrapping by Scenarios Created in USEd

In the actual implementation the sensors and actuator in client home are controlled by a Residential Gateway (RGW) which is connected to the Netcarity Server by a secure and authenticated channel, called Netcarity Secure Channel (NCSC). In the simulated case, the residential gateway is replaced by the SitCom run-time connected to Netcarity Server via the NCSC connection. There is no difference in behaviour or data types between simulated and real-live smart home from the Netcarity Server side. Multiple instances of SitCom run-times can simulate a whole network of smart homes and their residential gateways. This setup is currently used for tuning Netcarity Server's rule-based system, fall-detector and acoustic scene analysis modules, and for verification of proposed communication schemes. Beside the visual feedback during the development, the 3D visualization obviously serves also as a good demonstration platform.

7.3.3 Use Case Results

We have used USEd to create several artificial scenarios to test the behaviour of the whole Netcarity setup. Using simulated data to support fall-down scenario was crucial, as it is unlikely to record real fall in the real home. We have used semi-simulated approach in that case, i.e. using real recordings of human movements and actions combined with simulated falls. In addition, we have recordings of various activities of daily living and "forced" falls from lab environment (Libal et al., 2009).

8. Conclusions and Future Work

In this work, we introduced framework for visual analysis of user behaviour in ambient intelligence environment. The framework consists of five tasks that are performed during evaluation of the applications, depending on the use case. These tasks are divided into two groups. Testing of detection mechanism accuracy task and user scenario creation and editing task are tasks that belong to first group of tasks that focus on preparation of data for tasks in second group. Second group tasks focuses on visual analysis of user behaviour either in evaluation of single user scenarios task or in evaluation of cooperative scenarios task. The framework tasks are executed using two tools, SitCom and USEd. These tools have been exercised by many developers over the course of several years. The tooling framework is currently used by more than ten sites across Europe and it has been part of several technology demonstrations. To support the development within the Reference Architecture Model, we have implemented SitCom framework – an extensible Java toolkit to help in all phases of the development of the non-trivial life-cycle of context-aware services. We equipped SitCom with a set of functionalities that we found beneficial in development of perceptual applications:

- it simulates the environment and the perceptual input (manually created or recorded scenarios)
- it provides 2D and 3D visualizations of scenes, situations, and scenarios
- it works as a middleware between user services and the layer of perception (situation modelling)
- it supports the context model builders with a framework for data fusion and filtering
- it serves as an IDE for repetitive testing of context-aware services and applications by replaying recorded or simulated scenarios
- it provides portability between virtual and real devices

The SitCom environment was successfully applied during the whole development cycle of the CHIL platform and it is now helping to bootstrap the Netcarity system. In many ways, having an integrated tool specialized for context-aware applications has been helpful to identify the necessary pieces for the application, like the needed room information, specific sensor information, or the definition of the participants roles.
The USEd tool has been used to create, edit and tune user scenarios in ambient intelligence environment (especially in smart room). The tool proved to be not only usable add-on to the SitCom in cases of simulated scenarios creation and editing but also fast tool for transformation of video recordings into the user scenarios.

Usage of the framework and applications was demonstrated in three use cases, where a subset of tasks was used. Results of the use cases showed that both tools are usable and they are able to work with data and visualize them.

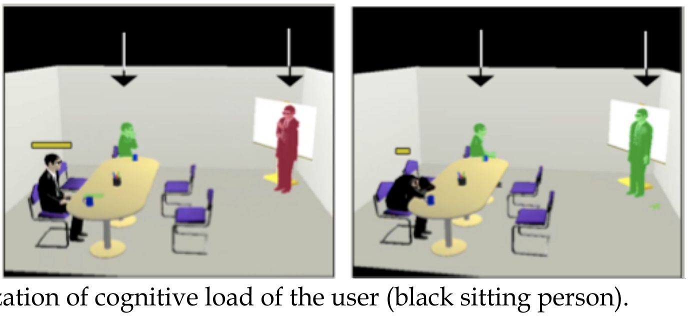
Fig. 14. Visualization of cognitive load of the user (black sitting person).

In the future work, we would like to focus on improvement of visualization in order to increase value of the visual analysis in evaluation of the ambient intelligence application. Currently, the visualization and data collection focus on behaviour of the user, like her/his position, her/his movement, head direction or detection of speech. We would like to add more information about the user interaction with objects in the environment. Such a type of visualization is closely linked up with the user model. This means that the advantage of this type of visualization is, that we can directly estimate the user cognitive load from the available data (user interaction log files). The cognitive load is rising, when the user is doing interactions, which are not expected by the developer, or the user is not sure how to continue to finish her/his task, and so on. By means of this visualization tool the evaluator gets instantly the global overview of the user performance (together with idea about critical points in the user interface investigated). The potential tool that can visualize behaviour of users (avatars) might generate the output given in Fig. 14. In such a way it is possible to investigate detail behaviour of individual users in complex situations. By means of colours it is possible to mark visually some specific features (parameters) of avatars that occurred in specific situations (in this case the cognitive load of the user in particular situations).

9. Acknowledgements

We would like to acknowledge support of this work by the European Commission under IST FP6 integrated project Netcarity, contract number IST-2006-045508. This work has also been partly supported by the Ministry of Education, Youth and Sports of the Czech Republic under the research program LC-06008 (Center for Computer Graphics) and the research program 6840770014 (Research in the Area of the Prospective Information and Navigation Technologies).

10. References

Barton, J. J. & Vijayaraghavan, V. (2002). UBIWISE, *A Ubiquitous Wireless Infrastructure Simulation Environment*. HP Labs Technical Report. HPL-2002-303.

Crowley, J. L.; Coutaz, J.; Rey, G. & Reignier. P. (2002). Perceptual components for context aware computing. In *Proceedings of UBICOMP 2002, International Conference on Ubiquitous Computing*, Goteborg, Sweden, pp. 117-134, September 2002.

Danninger, M.; Flaherty, G.; Bernardin, K.; Ekenel, H.; Köhler, T.; Malkin, R.; Stiefelhagen, R. & Waibel, A. (2005): The connector: facilitating context-aware communication. In: *7th Int. Conference on Multimodal Interfaces 2005*. ICMI'05, New York, USA, ACM Press. 2005.

Dey, A.; Salber, D. & Abowd. G. (2001). A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction (HCI) Journal*, Vol. 16, pp. 97-166, 2001.

Dimakis, N.; Soldatos, J. K.; Polymenakos, L.; Fleury, P.; Cuřín, J. & Kleindienst, J. (2008). Integrated Development of Context-Aware Applications in Smart Spaces, *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 71-79, Oct-Dec, 2008

Fleury, P.; Cuřín, J. & Kleindienst. J. (2007). SitCom - development platform for multimodal perceptual services. In *Proceedings of 3nd International Conference on Industrial Applications of Holonic and Multi-Agent Systems*, Regensburg, Germany, September 2007. V. Marik, V. Vyatkin, A.W. Colombo (Eds.): HoloMAS 2007, LNAI 4659, Springer-Verlag, pp. 104-113, 2007.

Gallud, J. A.; Lozano, M.; Tesoriero, R. & Penichet, V. M. R. (2007). Using Mobile Devices to Improve the Interactive Experience of Visitors in Art Museums. *In Human-Computer Interaction. Interaction Platforms and Techniques*, volume 4551 of LNCS, pp. 280–287. Springer, 2007.

Harrison, B. L.; Owen, R. & Baecker, R.M. (1994). Timelines: An Interactive System for the Collection of Visualization of Temporal Data. *In Proc. of Graphics Interface'94.* Toronto: Canadian lnfo. Proc. Society, pp. 141-148, 1994.

Ivory, M. Y. & Hearst, M. A. (2001). The state of the art in automating usability evaluation of user interfaces. *ACM Computer Survey*, Vol. 33(4), pp. 470-516, 2001.

i2home. The Intuitive Interaction for Everyone with Home Appliances based on Industry Standards (i2home) project. Web site (2009) http://www.i2home.org

Libal V.; Ramabhadran B.; Mana N.; Pianesi F.; Chippendale F.; Lanz O. & Potamianos G. (2009). Multimodal Classification of Activities of Daily Living in Smart Homes, *International Workshop of Ambient Assisted Living (IWANN 2009)*, S. Omatu et al. (Eds.), LNCS 5518, Springer, pp. 687-694, 2009.

Maly, I. & Slavik, P. (2007. Towards Visual Analysis of Usability Test Logs Using Task Models. In *Proc. of the 5th Workshop on Task Models and Diagrams 2006*. TAMODIA 2006. LNCS, vol. 4385, pp. 24–38. Springer, Heidelberg, 2007.

Mikovec, Z.; Maly, I.; Slavik, P. & Curin, J. (2007). Visualization of users' activities in a specific environment. In *Winter Simulation Conference (WSC) 2007*, pp. 738–746, Washington DC, USA, December 2007. WSC.

Nazari Shirehjini, A. A. & Klar, F. (2005). 3DSim: Rapid Prototyping Ambient Intelligence. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*. ACM International Conference Proceeding Series; Vol. 121, pp. 303-306, 2005.

Nazari Shirehjini, A. A. (2005). A Generic UPnP Architecture for Ambient Intelligence Meeting Rooms and a Control Point allowing for integrated 2D and 3D Interaction. In *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*. ACM International Conference Proceeding Series; Vol. 121, pp 207-212, 2005.

Nielsen, J. (1994). *Usability Engineering*. Morgan Kaufmann.

Nishikawa, H.; Yamamoto, S.; Tamai, M.; Nishigaki, K.; Kitani, T.; Shibata, N.; Yasumoto, K. & Ito, M. (2006). Ubireal: Realistic smartspace simulator for systematic testing. In *Ubicomp*, volume 4206 of LNCS, pp. 459–476. Springer, 2006.

Paterno, F.; Mancini, C. & Meniconi, S. (1997). Concurtasktrees: Diagrammatic notation for specifying task models. In *IFIP TC13 Sixth International Conference on Human-Computer Interaction*. pp. 362–369, Chapman and Hall, July 1997.

Paterno, F.; Russino, A.; Santoro & C. (2007). Remote Evaluation of Mobile Applications. In *Proc. of the 6th Workshop on Task Models and Diagrams 2007*. TAMODIA 2007. LNCS, vol. 4849, pp. 155–169. Springer, Heidelberg. 2007.

Rubin, J. (1994). Handbook of usability testing: *How to plan, design, and conduct effective tests*. John Wiley & Sons, Inc., New York, NY, USA, 1994.

Seo, J.; Goh, G. & Kim, G. J. (2005). Creating Ubiquitous Computing Simulators using P-VoT. In *Proceedings of the 4th international conference on Mobile and ubiquitous multimedia*, ACM International Conference Proceeding Series; Vol. 154, pp. 123-126, 2005.

Spence, R. (2000). *Information visualization*, ACM Press, Addison Wesley, 2000

Waibel, A. & Stiefelhagen, R. 2009. Computers in the Human Interaction Loop. *Human-Computer Interaction Series*, 2009, XXII, 374 p. 108 illus., 88 in color., Hardcover, Springer, Heidelberg, 2009.

Ambient Intelligence Interaction via Dialogue Systems

Porfírio Filipe [a, b, c] and Nuno Mamede [a, d]
[a] *L2F INESC-ID – Spoken Language Systems Lab*
[b] *GuIAA – Research Group on Autonomous Environments*
[c] *ISEL – Instituto Superior de Engenharia de Lisboa*
[d] *IST – Instituto Superior Técnico*
Portugal

1. Introduction

The vision of Ambient Intelligence (AmI), that expresses a paradigm in information technology, is based on the increasing technological advances in embedding computational power, information and sensing capabilities into everyday artefacts and environments (Ducatel et al., 2001). Intelligent environment is a technological concept that, according to Mark Weiser, is "*a physical world that is richly and invisibly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in the everyday objects of our lives, and connected through a continuous network*" (Weiser, 1991). Consequently, the computational model of this kind of environments can be analysed as a large collection of networked artefacts.

The use of embedded systems to control artefacts, tools and appliances has been common practice for almost two decades now. With every new generation, these controllers provide an ever increasing list of capabilities in the form of assistance, information, and customization. However, it is the addition of communication capabilities that changes the perspectives of what such systems can do: gather information from other sensors, real objects and computers on the network, or enable user-oriented customization and operations through short-range communication (Cook & Das, 2004).

For this, AmI technological infrastructures must be able to spontaneously reconfigure themselves and grow from the available, purposeful artefacts in order to become effective in the real world. Recent approaches are based on exploiting the affordances of real artefacts by augmenting their physical properties with the potential of computer based support (Streitz, 2007). Combining the best of both worlds requires an integration of real and virtual worlds resulting in hybrid worlds.

Despite the current availability of technology, there is a notorious absence of large scale settings. In this context, the fundamental question is what kind of intelligent interfaces is needed to access a large federation of artefacts within an AmI scenario. In this scenario, each one of the hybrid artefacts or controllable resources should be designed to allow plug and play integration in an AmI multimodal architecture (Dahl et al., 2008).

According to the convincing demonstration of Byron Reeves and Clifford Nass, the interactions with computers, television, and new communication technologies are identical to real social relationships and to the navigation of real physical spaces (Reeves & Nass, 1996). In this perspective, it is reasonable to assume, for instance, that people would talk naturally with a household appliance.

The design of natural language applications that allow people to talk with machines or computers, in the same way that they talk with each other, is materialized under the form of a Spoken Dialogue System (SDS) (Zue & Glass, 2000; McTear, 2004), having constituted a natural interface, where the use of speech is privileged.

Currently, it is unrealistic to consider a real existence of an autonomic SDS embedded into each one of the environment's artefacts, because of real hardware limitations. Nevertheless, the coordination and collaboration between a set of autonomic SDS is, per se, a huge challenge.

In order to achieve AmI interaction via SDS (Minker et al., 2009) each controllable resource should implement, at least, an adequate semantic interface to expose the resource's functional capabilities at knowledge level (Newell, 1982). Nevertheless, this semantic interface is not only for exclusive use of the SDS because it can be also freely used by other concurrent systems.

Essentially, a semantic interface is a collection of task descriptors used to expose artefact's functional capabilities, and is sustained by a set of concepts that are atomic knowledge units. The mining of a concept can be previously established or can be dynamically inferred comparing its linguistic knowledge or its semantic references to internal or external knowledge source nodes (Filipe & Mamede, 2008; Filipe & Mamede 2009). This approach focus on the representation and management of the environment's knowledge aims to satisfy dialogue management needs related to the environment semantic interoperability. For this, an environment Knowledge Aggregation Process (KAP) built-in a distributed Environment Interaction Manager (EIM) manages federations of resources within an AmI holistic vision (Aartsa & Ruytera, 2009).

Summarizing, our contribution enables spontaneous reconfiguration of SDS, within an AmI holistic vision, to provide speech natural interaction. Section 2 gives an overview of the state of the art in SDS. Section 3 presents relevant issues about spontaneous portability.

Section 4 gives an overview of the knowledge modelling approach for semantic interface design. Section 5 describes the Knowledge Aggregation Process (KAP). Section 6 presents the experimental setup describing practical issues as well. Section 7 summarizes the contribution's main topics, conclusions and future work.

2. Spoken Dialogue Systems

The origins of SDS can be traced back to Artificial Intelligence (AI) research in the 1950s concerned with developing conversational interfaces. The research of SDS is commonly considered a branch of human-computer interaction, although its origins are generally rooted in the automatic speech recognition community.

However, it is only within the last decade or so, with major advances in speech technology, that large scale working systems have been developed and, in some cases, introduced into commercial environments. The integration of components into a working system is still an important key issue (McTear, 2004).

Typically, a SDS is used to access the data source of the domain often materialized under a relational database. The traditional interaction cycle starts when the user's request, which is captured by a microphone, provides the input for the Speech Recognition component. Next, the Language Understanding component receives the recognized words and builds the related speech acts. The Dialogue Manager (DM) processes the speech acts, accesses the Data Source and then calls the Response Generation component to generate a response message for the user. Finally, the message is processed by the Speech Output component to produce speech. The response of the SDS can be final or a request for clarification. When everything is acceptable, a final answer is produced based on the obtained external data.
Fig. 1 shows a typical logical flow through SDS components architecture to access the domain data source, typically sustained by a relational database.

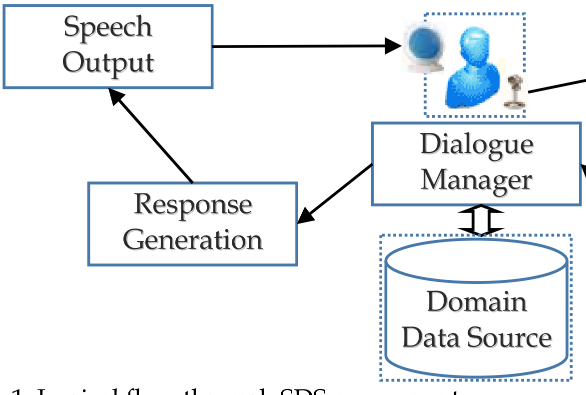

Fig. 1. Logical flow through SDS components

Current trends are putting more research emphasis on aspects of psychology and linguistics. Speech-based human-computer interaction faces several challenges in order to be more widely accepted. One of these challenges is the domain portability.

3. Spontaneous Portability

This section describes spontaneous domain portability issues, focusing a dynamic domain interaction within an AmI vision.
The design of a DM can be customized to new domains in which different dialogue strategies can be explored, concerning only phenomena related to the dialogue with the user, focusing on dialogue and on discourse strategies. The DM component should not be involved in the process of accessing a background system or performing domain reasoning.
This divide to conquer approach assumes that practical dialogue and domain-independent hypothesis are true (Allen et al., 2000). The reason is that typical applications of human computer interaction involve dialogue focussed on accomplishing some specific task. Assuming this, the bulk of the complexity in the language interpretation and dialogue management is independent of the task being performed. In this context, a clear separation between linguistic dependent, and domain dependent knowledge, is needed for reducing the complexity of SDS typical components, specially the DM.

For this, should be considered another component of SDS architecture, which handles these features, namely a Domain Knowledge Manager (Flycht-Eriksson, 2000). This component is in charge for retrieving and coordinating knowledge from the different domain knowledge sources and application systems, traditionally named background system. In these circumstances, this approach allows the customization of the DM enabling domain portability and easy configuration of the SDS architecture.
However, in this paper, the SDS is seen as a computational entity that allows universal access to AmI. In this interaction scenario, the SDS should be a computational entity that allows access to any resource by anyone, anywhere, at anytime, through any media or language, allowing its users to focus on the task, not on the tool.
Nevertheless, a traditional SDS cannot be directly used to interact with an intelligent environment, due lack of spontaneous portability, because of the fact that SDS are not ubiquitous yet. Within a ubiquitous domain, one does not know, at design time, all the resources that will be available. To address this issue, an approach for SDS architecture improvements and knowledge modelling for semantic interface design is needed (Filipe, 2007; Filipe & Mamede, 2008; Filipe & Mamede, 2009).
The SDS customization for AmI access, allowing spontaneous configuration, is supported by the proposed Environment Interaction Manager (EIM) (see Fig. 2).

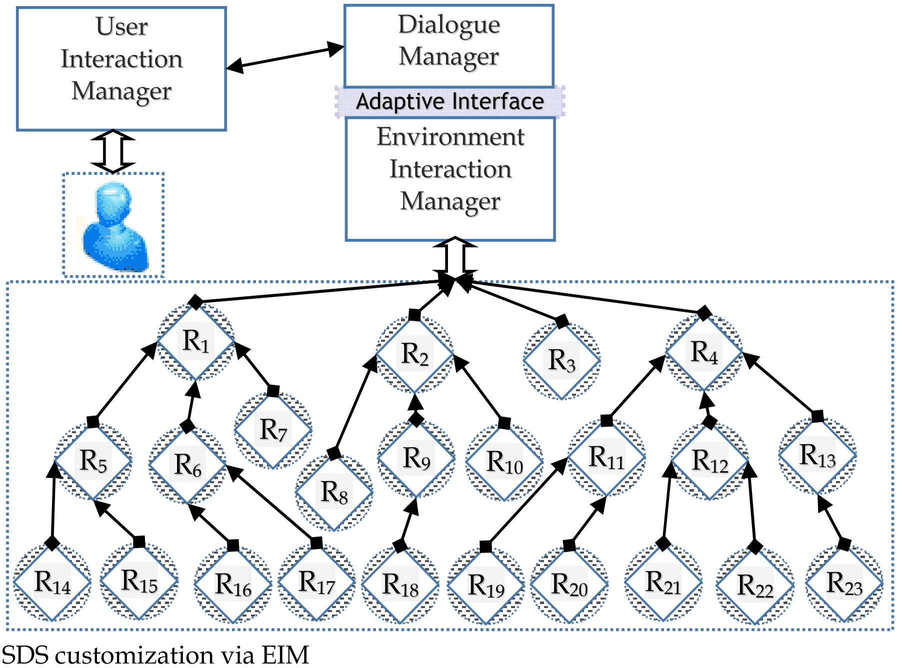

Fig. 2. SDS customization via EIM

The main goal of the EIM is to support the communication interoperability between the SDS and a set of controllable resources, performing the environment's knowledge management for allowing spontaneous configuration. For this, the EIM includes a knowledge model (see Section 4) that represents all the aggregated resource's semantic interfaces.

For instance, when it refers to an indoor environment, the knowledge reflects the plan or physical organization of the building and the SDS controllable resources. The building is modelled as a spontaneous aggregation "part-whole" of controllable resources. Each resource shares a semantic interface that exposes its functional capabilities and makes possible its manipulation by the SDS. The building itself is seen as a large controllable resource that aggregates minor resources, such as floors, rooms, entrance halls, foyers. Each one of these resources can aggregate other resources that control, for instance, doors, windows, elevators, environment controls, multimedia controls, appliance controls and so on.
When a resource, designated by "part", is activated, a discovery protocol searches the nearest resource, designated by "whole". After that, KAP is executed, in the context of the "whole", merging the knowledge built in the semantic interface of the "part" (to the "whole" knowledge model) and propagates the changes to related building parts.
In order to allow a large federation of resources, managed by the EIM, several aggregation levels are considered (see Fig. 2). A controllable resource is distinguished by a different identifier ($R_1 \dots R_n$) and can be aggregated to another resource belonging to an upper level or directly to EIM. The last aggregation level, the level of the EIM, holds all the existing resources semantic interfaces that can represent, for instance, an entire building.

3.1 Adaptive Interface

In order to allow flexible behaviour in SDS interaction, the DM must be able to handle under specified requests (when user provides only partial information). The DM must decide which task should be performed. Our approach recommends the use of domain dependent knowledge, for instance, in a particular domain the user's preferred choice may be "*turn-on the light*" for the request "*turn-on*", but in another domain without lights, the best choice is certainly different.
Since the DM must not know the EIM's domain model, the DM must submit requests to be answered by EIM. For this, an adaptive and easy to use EIM's interface is needed (Filipe et al., 2007; Filipe et al.; 2008). The EIM presents to the Dialogue Manager (DM) an aggregated view of the environment. In this architecture (see Fig. 2), a clear separation is assumed between discourse or dialogue dependent issues (DM job) and domain dependent issues (EIM job).
The ideas behind this adaptive interface are based on the relative weight (relevance) of each concept, which is included in the request. Two independent ranking, for tasks and resources, are used to compute the best task-resource pairs. The interfaces accept as input a list of pivot concepts. The concepts reference tasks and resources, which are translated to points credited in the respective or task or resource rank.
In some cases, this ranking algorithm conduces to situations where the tasks or resources have similar or even the same ranking position. In this situation, the best task-resource pair is not clearly determined, demanding for large clarification dialogues.
One way to address this issue is by endowing the DM with the ability to interactively learn dialogue strategies, namely by using reinforcement learning approaches (Henderson et al., 2005; Schatzmann et al., 2006).
However, although this kind of approaches present interesting characteristics in what concerns learning of dialog strategies, they suffer from well-known drawbacks for online operation, especially in what concerns the number of interactions needed for convergence,

which restricts their application mainly to offline processing. On contrary, in online dynamic environments, the user interactions are relatively scarce and the SDS must be able to adapt its operation taking advantage of these limited interactions.

The main focus of the adaptive interface is not on learning complex dialogue strategies, but on dynamically adapting the relevance of task-resource pairs according to user interaction, watching the selection or rejection expressed in previous user's clarification dialogues. As an example, let's consider the case where the user is in the kitchen and selects the task "turn-on". Since "turn-on" must refer to some resource, SDS asks the user to specify the resource he/she wants to refer (e.g., a microwave oven, or the ceiling lights). Therefore, the DM must decide about which resources it should ask the user first. Consequently, some form of resource relevance is needed to enable the selection of resources according to the selected task and the previous history of user's interaction.

To allow for a dynamic adaptation considering the relevance of resources, a simple implementation of the activation potential model should be made, following the Agent Flow Model proposed by Morgado and Gaspar (Morgado & Gaspar, 2004).

4. Semantic Interface

This section gives an overview of the most relevant components of the knowledge model that holds the design of the semantic interface, which includes four independent knowledge components: the discourse model, the task model, the world model, and the events model.

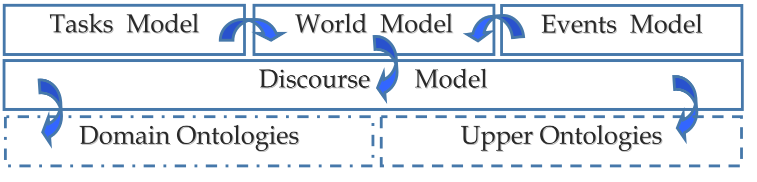

Fig. 3. Semantic Interface Knowledge Model

Additionally, external ontological knowledge components are also considered to allow the integration with domain ontologies and upper ontologies (see Fig. 3).

4.1 Discourse Model

The discourse model defines a conceptual support, grouping concept descriptions, used essentially to express functional resource capabilities. The mining of a concept is previously established or is inferred at runtime by KAP, comparing its linguistic descriptors or its semantic descriptors. For instance, a semantic descriptor can include a Uniform Resource Identifier (URI) that points to external ontology nodes.

A concept descriptor defines an atomic unit of knowledge and maps linguistic knowledge into domain knowledge. Essentially, a concept maps a set of URIs into a set of terms or more generically into a set of Multi-Word Unit (MWU). Concepts declarations include linguistic and semantic parts organized according to main kinds, which are: "task", "role", "event", "name", and "constant". Concepts of kinds "action" or "perception" hold task names. A perception task cannot modify the state of the environment, whereas, an action task can. Concepts of kinds "collection" or "quantity" hold task roles (parameters or arguments). The

kind "collection" is used to define sets of constants (represented also by concepts such as white, black, red, …) to fill task roles (colour, shape, texture, …). The kind "quantity" is about numbers (integer, real, positive, …) and the "unit" kind is for measures (time, power, …). The kind "event" holds event names. The kind "name" holds resource or class names.
In order to ensure the availability of vocabulary to refer the represented concepts, concept descriptions include linguistic properties. Each Word (or term), has a part of speech tag, such as noun, adjective, verb, adverb; a language tag, such as "pt-PT", "pt-BR", "en-UK" or "en-US"; and an optional phonetic transcription. The linguistic description holds a list of words, or more generically a MWU, referring linguistic variations associated with the concept, such as synonyms, acronyms and even antonyms.
Fig. 4 shows a mind map of a concept descriptor.

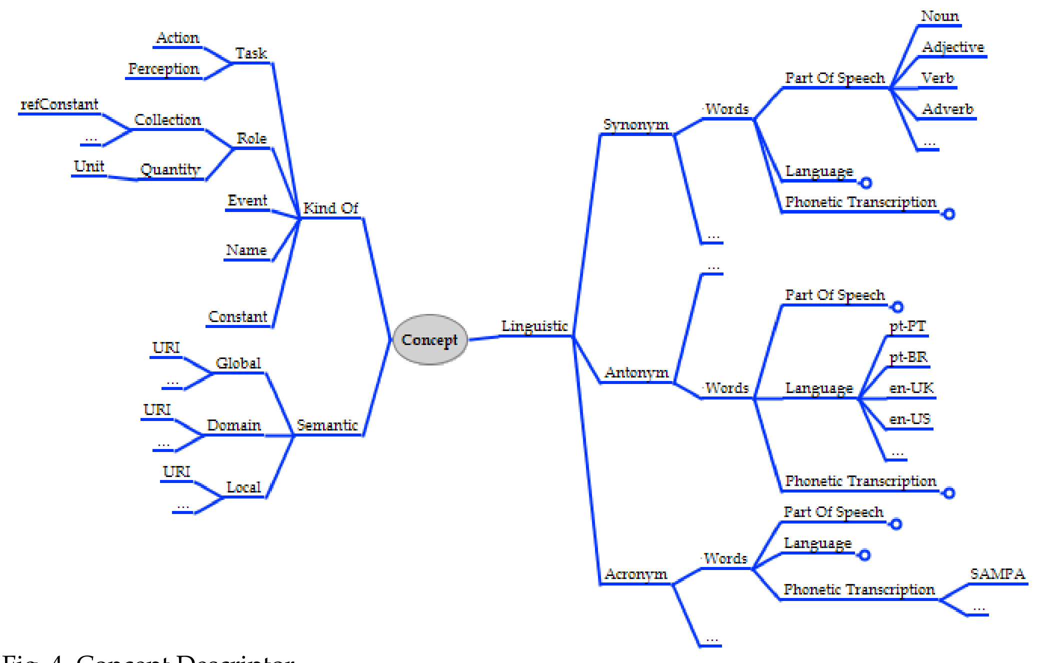

Fig. 4. Concept Descriptor

Concept descriptors can also hold semantic references typically characterized by a Universal Resource Identifier (URI). The semantic description supports references to domain knowledge sources (domain hierarchy, domain ontologies) or global knowledge sources, (upper ontologies or a lexical database, such as WordNet). A concept description must include at least one URI for local reference.
The references about knowledge sources must be unique in the same knowledge model and must be encoded using a particular data format to allow a unique identification of the referenced concept. The syntax of the knowledge source references does not need to be universal; it is enough that each particular knowledge source shares the same syntax.

4.2 Task model

The task model contains one or more task descriptions, based on concepts previously declared in the discourse model. Fig. 5 shows a mind map of a task descriptor.

A task descriptor is a semantic representation of a task that holds a task name and, optionally, a role input and/or output list. A role describes an input and/or output task argument or parameter. An input role has a name, a range, and a restriction.

The role restriction is a rule that is implemented as a regular expression and is optional. An output role is similar to an input role with an optional default constant.

The initial and final rules perform environment state validation: the initial rule (to check the initial state of the world before a task execution) and the final rule (to check the final state of the word after a task execution). These rules, also implemented as regular expressions, can refer to role names and constants returned by perception task calls.

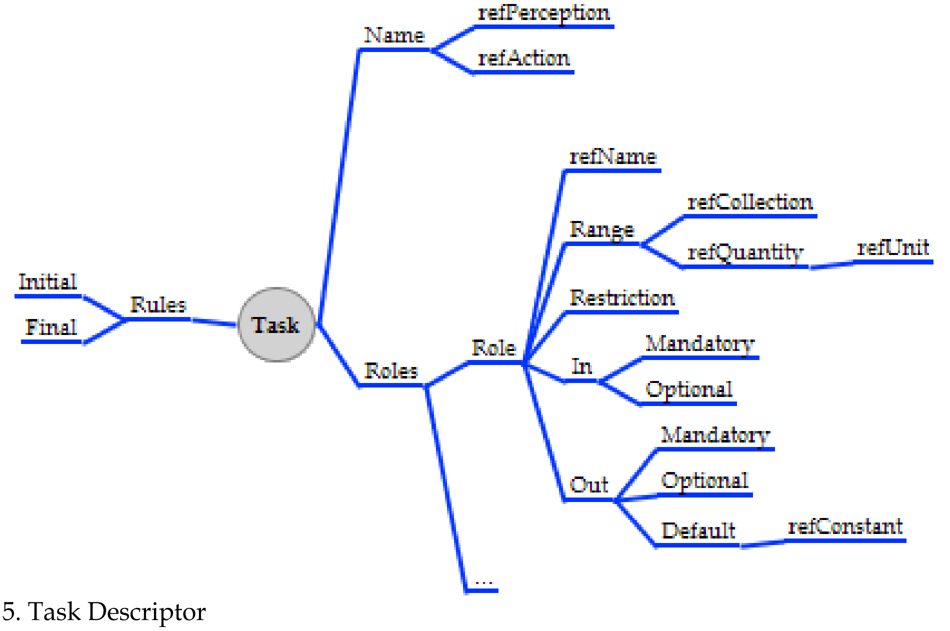

Fig. 5. Task Descriptor

4.3 World Model

The world model contains descriptions about one or more resources (the "whole" and its "parts") properties. These descriptions refer mandatorily to concepts previously declared in discourse model including, for instance, name, and optionally physical properties (colour, shape, …) that are known by the SDS user and are typically used to indentify or select a resource within a user request. Optionally, a resource description refers to one or more classes symbolized in the domain ontologies.

Fig. 6 shows a mind map of a resource descriptor.

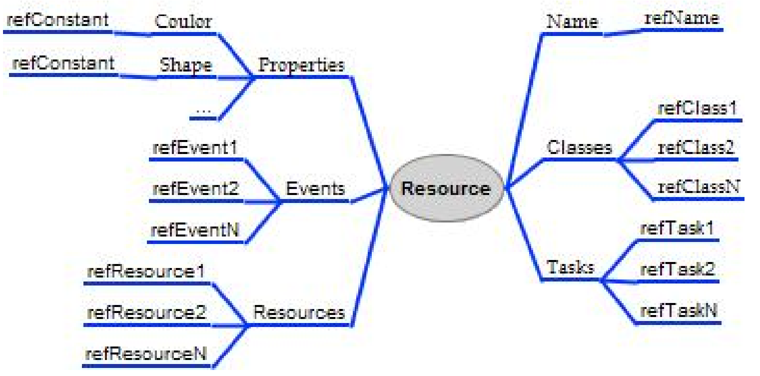

Fig. 6. Resource Descriptor

4.4 Events Model

The events model contains descriptions of events supported by concepts also declared in the discourse model. These descriptions are similar to task descriptions only with name and input roles. An event is a notification about an expected or unexpected environment state modification. The events model supports the reactive behaviour of the EIM and is used to notify the DM of the SDS about the environment changes.

5. Knowledge Aggregation Process

The main goal of the Knowledge Aggregation Process (KAP) is to update on-the-fly the knowledge model of a resource semantic interface "whole", merging the knowledge originated by one "part", that is also a resource. Generally, it is assumed that each resource holds its own built-in semantic interface. However, due to hardware limitations, the semantic interface can be maintained and virtualized by other computational entity.

At its starting point, KAP puts side by side concepts and tasks descriptions using similarity criteria:

(a) Two concepts are similar when its domain or global URIs is the same or its linguistic descriptors are literally equal. When the kind of the concepts is "collection", its member constants must be also similar;

(b) Two tasks are similar when its descriptions are literally equal;

In order to update the knowledge model of a "whole", KAP follows the next four steps:

(1) For each concept in "part", without a similar (a) in "whole", is added a new concept description to "whole" discourse model;

(2) For each task in "part" without a similar (b) in "whole" is added a new concept description to "whole" task model;

(3) A new resource description is added to "whole" world model;

(4) The resource description is linked to the updated tasks descriptions.

6. Experimental Setup

The experimental setup is based on our simulator, originally developed for Portuguese users. Fig. 7 shows a screen with a summary of the current knowledge model.

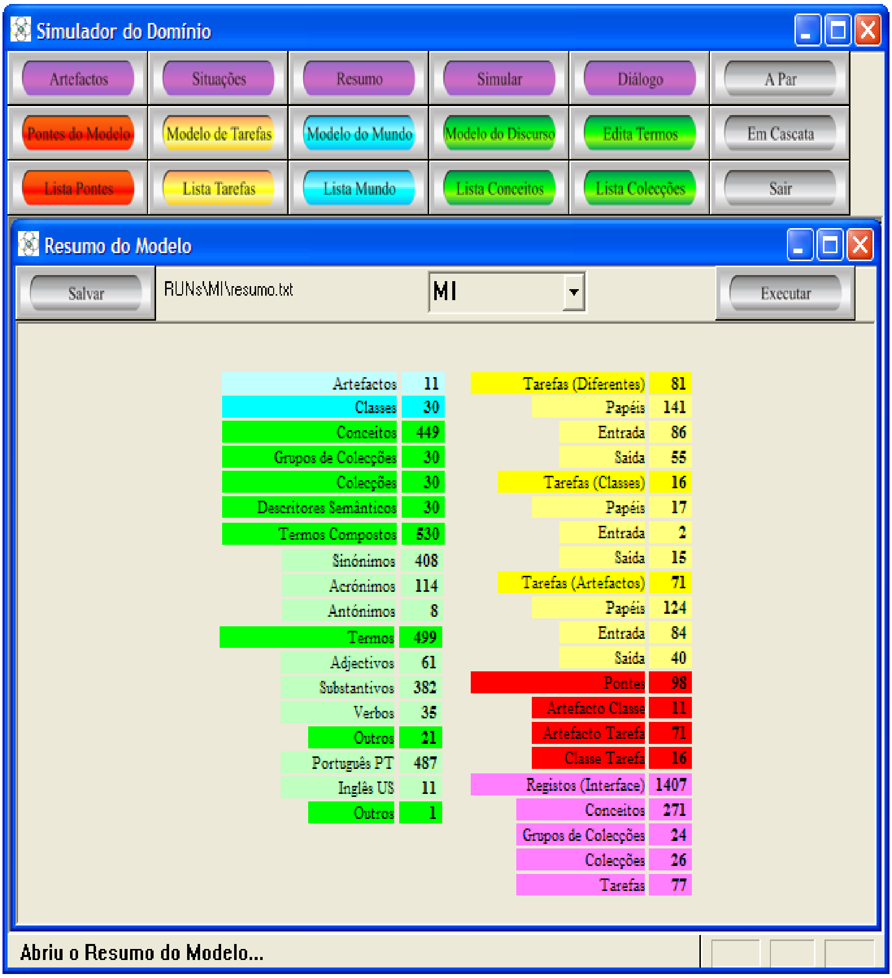

Fig. 7. Screen of the simulator with the summary of the model

The simulator incorporates an elementary dialogue manager and allows the debug of an invoked task analyzing the interaction with the target resource. Is possible to debug KAP execution, execute tasks, and observe its effects on the environment. We can also consult and print several data about each resource semantic interface. Currently, the simulator holds approximately a total of one thousand concepts and one hundred tasks.
It is possible to query the simulator about detailed descriptions of the represented concepts.
Fig. 8 shows a screen of the simulator with the description (English) of the microwave oven concept in the current knowledge model, according to its definition in WordNet (Fellbaum, 1998).

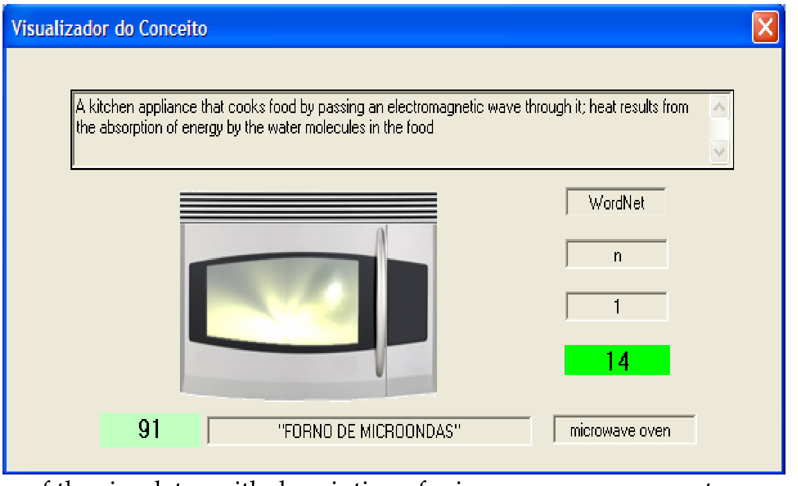

Fig. 8. Screen of the simulator with description of microwave oven concept

Considering that the environment is characterized by an arbitrary set of resources, physically supported by augmented artefacts, such as appliances, furniture, or ambient controls the simulator makes available several autonomic resource simulators, such as air conditioning, freezer, fryer, light source, microwave oven, table, water faucet, window, and window blind.
Fig. 9 shows part of the class hierarchy of resources available in the simulator.

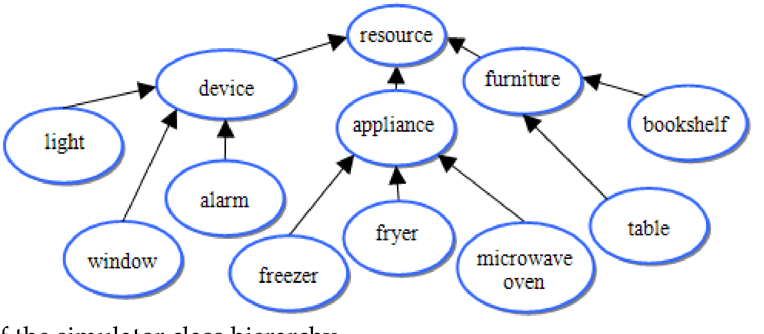

Fig. 9. Part of the simulator class hierarchy

The class hierarchy does not need to be complete because it can be improved as new resources are dynamically added.
These resources are activated to simulate and intelligent environment, composed by several floors and rooms defining aggregation levels that can be used, for instance, to model dwellings. A room is modelled as a resource that aggregates other resources physically present in that room, such as kitchen, living room, dinning room, and bedroom. A floor is a resource, which aggregates its physical parts generically named by rooms. At top level of the simulation, EIM includes a knowledge representation of the entire building.
For instance, when the SDS user demands "*turning on the light*", the simulator of the light source, aggregated by KAP as part of the kitchen simulator, executes the request turning on the kitchen light indicating by default thirty percent of luminosity.

Fig. 10 shows the screen of the kitchen light simulator, with its properties and state data, after the execution of the request "*turning on the light*".

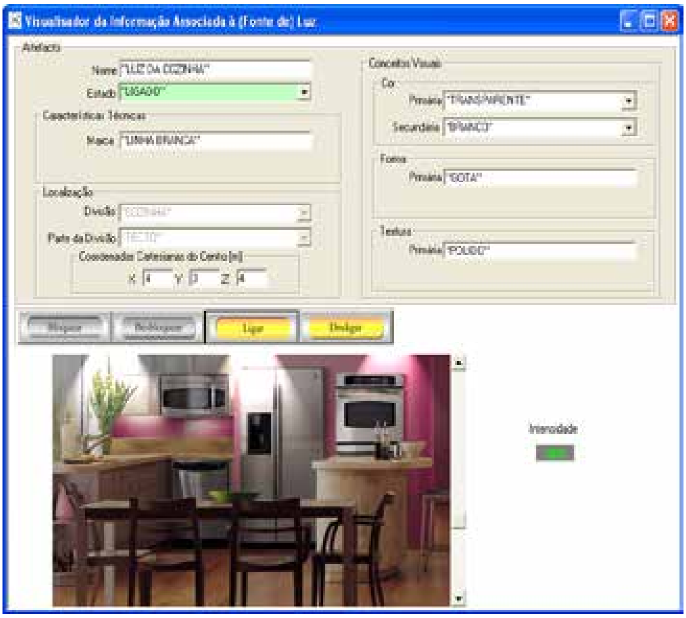

Fig. 10. Screen of the kitchen light simulator

For instance, when the SDS user demands "defrosting hamburger", the simulator of the microwave oven, aggregated by KAP as part of the kitchen simulator, executes the request indicating the automatically select power (300 watts – see symbol) and duration (8 minutes) of the defrosting process.

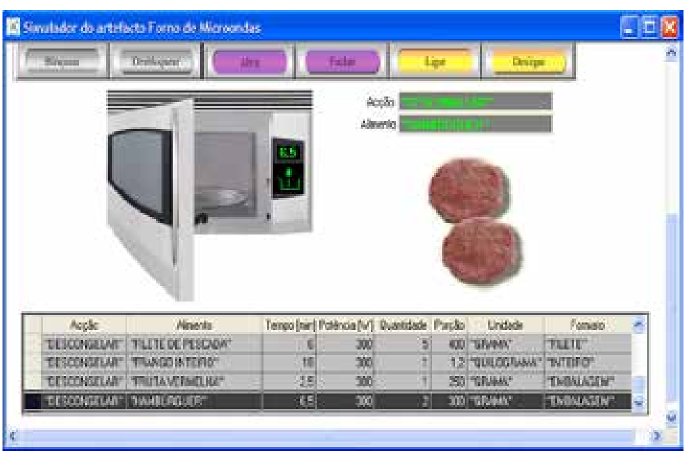

Fig. 11. Screen of the microwave oven simulator

Fig. 11 shows the state of the microwave oven simulator after the execution of the request "*defrosting hamburger*".

In order to start the defrosting process, after closing the door of the microwave, the user must confirm its previous intention indicating the request "*turning on the microwave*".
Fig. 12 shows the screen of the fryer simulator after the execution of the request: "*frying Chinese spring rolls*". This screen shows the automatically select temperature (180 ºC) and duration (7 minutes) of the frying process.

Fig. 12. Screen of the deep fryer simulator

Fig. 13 shows the screen of the freezer simulator after the execution of the request "*what is the amount of carrots?*". The table in the simulator shows the selected type of food. However, in this case the dialogue manager also returns the answer "*1 package with 300 g*".

Fig. 13. Screen of the simulator of the freezer simulator

The simulator includes also the treatment of more complex user's requests, for instance, involving relational operators. For example, the request "*what is the food with amount less than five*" is redirected to the freezer simulator, by the kitchen simulator, producing the list of food with amount less than five.
The concept "carrot" (in Portuguese "cenoura") is included in the microwave oven and in the freezer simulators interfaces. However, the EIM has only one definition of the concept "carrot" in its knowledge model automatically declared by KAP at runtime.

7. Concluding Remarks

Current technologies require human intervention to solve environment reconfiguration problems. The growth in pervasive computing will require standards in real objects interoperability to achieve AmI vision. In order to face this issue, a more human like way of interaction is proposed including spoken natural language support within intelligent environments. Our proposal tries to improve the configuration features of the SDS architectures with a semantic-based approach allowing an autonomic design of semantic interfaces, which are used to describe resource capabilities. For this, was proposed EIM (Environment Interaction Manager) SDS component and KAP (Knowledge Aggregation Process) to deal, at runtime, with federations of resources. In this context, the computational environment can handle completely new resources of unknown or unseen classes, trying to cover the ubiquitous essence of natural language.

The presented ideas have been applied with success implementing the spontaneous configuration of knowledge-based resources. The proposed semantic-based approach, supported in the field by the EIM, is a significant contribution to improve the portability, scalability and, simultaneously, the robustness of the SDS being developed in our lab.

Currently, our work is based in the kitchen environment. However, we intend to generalize the use of the SDS natural interface to support inhabitants' activities, for instance, to optimize climate and light controls, item tracking and general use items, automated alarm schedules to match inhabitants' preferences, and control of media systems.

In the near future, we aim to study more deeply the knowledge replication versus knowledge integration rate. We expect to prove, for the upper aggregations levels, an interesting knowledge integration rate, due to the reuse of similar concepts and tasks within the same intelligent environment.

8. References

Aartsa, E. & Ruytera, B. (2009). New research perspectives on Ambient Intelligence, *Journal of Ambient Intelligence and Smart Environments,* Vol. 1, pp. 5–14

Allen, J.; Byron, D.; Dzikovska, M.; Ferguson, G.; Galescu, L. & Stent, A. (2000). An Architecture for a Generic Dialogue Shell, *Natural Language Engineering*, Vol. 6, 3-4, pp. 213-228, Cambridge University Press

Cook, D. & Das, S. (2004). *Smart Environments: Technology, Protocols and Applications,* Wiley Interscience, ISBN: 978-0-471-54448-7

Dahl, D.; Kliche, I.; Tumuluri, R.; Yudkowsky, M.; Bodell, M.; Porter, B.; Raggett, D.; Raman, T. & Wahbe, A. (2008). Multimodal Architecture and Interfaces, *http://www.w3.org/TR/mmi-arch/*

Ducatel, K.; Bogdanowicz, M.; Scapolo, F.; Leijten, J. & Burgelman J. (2001). Scenarios for Ambient Intelligence in 2010, *ISTAG: European Commission Report*

Dzikovska, M.; Allen, J. & Swift, M. (2003). Integrating Linguistic and Domain Knowledge for Spoken Dialogue Systems in Multiple Domains. *Proceedings of Workshop on Knowledge and Reasoning in Practical Dialogue Systems at The Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-2003)*, pp. 25-35, Acapulco, Mexico, August 2003

Fellbaum, C. (Ed.) (1998). *WordNet, an Electronic Lexical Database*, MIT Press, Cambridge, MA, United States

Filipe, P. & Mamede, N. (2008). Empirical Multi-Artifact Knowledge Modeling for Dialogue Systems, *Proceedings of 11th International Conference on Enterprise Information Systems*, pp. 286-292, Barcelona Spain, June 2008

Filipe, P. & Mamede, N. (2009). Indoor Domain Model for Dialogue Systems, *Proceedings of 13th International Conference on Human-Computer Interaction "Universal Access in HCI, Part III, LNCS 5616"*, pp. 512-520, ISBN: 978-3-642-02712-3, San Diego CA USA, July 2009, Springer-Verlag, Berlin, Heidelberg

Filipe, P. (2007). *Dynamic Integration of Artefacts in Dialogue Systems*. PhD Thesis, Technical University of Lisbon UTL/IST

Filipe, P.; Araújo, P. & Mamede, N. (2008). Dialogue Systems Domain Interaction Using Reinforcement Learning, *Proceedings of 11th Ibero-American Conference on Artificial Intelligence (IBERAMIA) - Workshop on Agreement Technologies (WAT 2008)*, Lisboa, Portugal

Filipe, P.; Morgado, L. & Mamede, N. (2007). An Adaptive Domain Knowledge Manager for Dialogue Systems, *Proceedings of 9th International Conference on Enterprise Information Systems (ICEIS 2007)*, pp. 45-52, Funchal, Portugal

Flycht-Eriksson, A. (2000). A Domain Knowledge Manager for Dialogue Systems, *Proceedings of 14th European Conference on Artificial Intelligence (ECAI 2000)*, IOS Press, Amsterdam, Amsterdam

Henderson, J., Lemon, O., Georgila, K. (2005). Hybrid Reinforcement/Supervised Learning for Dialogue Policies from Communicator Data. Proceedings of 19th International Joint Conference on Artificial Intelligence (IJCAI'05) Workshop on KRPDS, Edinburgh, Scotland

McTear, M. (2004). *Spoken Dialogue Technology: Towards the Conversational User Interface,* Springer-Verlag, ISBN 1-85233-672-2, London, Berlin, Heidelberg

Minker, W.; López-Cózar, R. & McTear, M. (2009). The Role of Spoken Language Dialogue Interaction in Intelligent Environments, *Journal of Ambient Intelligence and Smart Environments,* Vol. 1

Morgado, L., Gaspar, G. (2004). Focusing Reasoning Through Emotional Mechanisms. *Proceedings of 16th European Conference on Artificial Intelligence (ECAI 2004)*. IOS Press. Valencia, Spain.

Newell, A. (1982). The Knowledge Level, *Artificial Intelligence*, Vol. 18, No. 1, pp. 87-127

Reeves, B. & Nass, C. (1996). *The Media Equation: How People Treat Computers, Television and New Media Like Real People and Places*, Cambridge University Press, ISBN: 157586052X, Cambridge, Mass

Schatzmann, J., Weilhammer, K., Stuttle, M., Young, S. (2006). A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies, *The Knowledge Engineering*, Vol. 21, No. 2, pp. 97-126

Streitz, N.; Kameas, A. & Mavrommati, I. (Eds.)(2007). *The Disappearing Computer: Interaction Design, System Infrastructures and Applications for Smart Environments,* Springer Publishers, ISBN: 3-540-72725-6, Berlin, Heidelberg, New York

Weiser, M. (1991). The Computer of the Twenty-First Century. *Scientific American,* Vol. 265, No. 3, pp. 94-104

Zue, V. & Glass J. (2000). Conversational Interfaces: Advances and Challenges. *IEEE,* Vol. 88, No. 8, pp. 1166-1180

6

Trust in global computing systems as a limit property emerging from short range random interactions*

Vasia Liagkou and Paul Spirakis
University of Patras, Department of computer Engineering and Research and Academic Computer Technology Institute
26500, Rio, Patras, Greece

Yannis C. Stamatiou
University of Ioannina, Mathematics Department
451 10, Ioannina, Greece

Effie Makri
University of the Aegean, Department of Mathematics
Greece

Abstract

Today we are witnessing a major reconsideration of the computing paradigm, as evidenced by the abundance and increasing frequency of use of novel terms such as *ambient intelligence*, *ubiquitous computing*, *disappearing computer*, *grid computer*, *global computing* and *mobile ad-hoc networks*. Systems that can be described with such terms are of a dynamic, with no clear physical boundary, nature and it seems that it is impossible (or, at least, difficult) to define sharply a number of important properties holding with certainty and throughout the whole lifetime of the system. In this chapter we propose a new paradigm for the concept of *trust* that can be applicable to describing trust related properties in evolving, "boundary-transcending", computing systems. This paradigm is founded on the interaction between *formal logic* and *threshold phenomena*, i.e. properties of large combinatorial structures that can be proved to emerge with certainty, as the system evolves. We define a number of notions of trust within these frameworks and pinpoint their inherent weakness in providing clear and measurable trust properties. We then argue that trust in dynamic, global computing systems must, necessarily, incorporate, to some degree, some non-formalizable elements, such as common sense and intuition in order to overcome formalism's weaknesses and result in a pragmatic notion of trust applicable to today's new computing paradigms.

*This work has been partially supported by the ICT Programme of the European Union under contract number ICT-2008-215270 (FRONTS) and by the Open University of Cyprus within the Programme (DYSAT).

1. Introduction

Although it is rather straightforward to assert that we trust a person stating, at the same time, the reason behind our attitude towards the person (e.g. good previous collaboration, absence of hostile moves from this person etc.), it seems very difficult to come to a conclusive trust statement when we confront a *device* (e.g. mobile phone, wireless computing node, company server etc.) which we must use in order to perform a necessary task (e.g. but a book on-line). Although in a sufficiently large interconnected system, like the nodes composing an sensor network, all pairs of entities are only a few communication hops apart, and thus if we cannot assert trust towards an entity, some other entity we trust could be of help, there are two major obstacles to the applicability of this approach: i) trust does not seem to possess "nice" logical properties that can support formal deduction processes like, for instance, the transitivity property implied above, and ii) decisions as to whether we should trust an entity often have to be made within an infinitesimal time interval (for instance, when an electronic transaction is pending and needs to be completed soon) and, thus, automation in trust manipulation is a highly desirable property of any formalization of the trust concept.

It is true that *trust* is a notoriously difficult concept when applied to machines and general computing systems. Many attempts towards a viable definition are primarily based on the intuition as to what are the desirable and non-desirable properties of a specific target system. Such a definition seems to be especially difficult to apply within the realm of the *new computing paradigm* that seems to have evolved over the past few years. This paradigm is a result of technological advances that made possible the construction of inexpensive, small and equipped with wireless communications capabilities computing devices which are able to form large, "shapeless", boundless, *global* computing systems. As difficult as it is to define, trust, nevertheless, plays a major role in the viability and usability of such a complex system. The most interesting areas of ubiquitous systems' security include trust management problems. Accountability and trust management pose new research problems because of the transient and decentralized nature of typical ubiquitous systems. Moreover those systems require interaction between large numbers of different unknown indemnities. A global trust evaluation model becomes necessary in this situation to determine the trust for each other and it must provide a computational representation of trust.

There is much ongoing research on the development and analysis of new trust management models for complex and dependable computer systems. Blaze *et al.* in (Blaze et al., 1996) proposed the application of automated trust mechanisms in distributed systems. Josang (Josang, 1996) focuses on the strong relationship between the issue of trust and the security concepts. Moreover a number of schemes for the design of a secure computer framework have been proposed (see (Eschenauer et al., 2002), (Hubaux et al., 2001)) which are based on automated trust management protocols. The propagation and composition of trust information is of pivotal research interest and many research papers (see (Guha et al., 2004; Kamvar et al., 2003; Richardson et al., 2003; Theodorakopoulos & Baras, 2004)) have proposed solutions. Grandison and Sloman try to see the trust as a belief (Grandison & Sloman, 2000). Based on a brief analysis they formulate the trust as *a firm belief in the competence of an entity to act dependably, securely and reliably within a specified context*. Moreover they establish the trust as a composition of several different attributes - such as reliability, dependability, honesty, truthfulness, security, competence, and timeliness - which may have to be considered depending on the environment in which trust is being specified. Here we take a different direction, we follow Dimitrakos' (see (Dimitrakos, 2001; Dimitrakos & Bicarregui, 2001)) definition of trust. We believe that *the trust of a party A in a party B is the measurable belief of A in B behaving dependably for* *a specified period within a specified context in relation to X*. Here we define the trust for a service *X* as a service requestor *A* to a service provider B for a service *X*. Thus, *A* and *B* are interlinked with a trust relationship, directed from *A* to *B*.

In this chapter, we provide some considerations and questions as to the extend to which trust can be mechanized and be based on *formally* definable properties that hold, almost certainly, in the limit in randomly growing combinatorial structures that model "shapeless" computing systems (e.g. dynamic ambient intelligence networks). We draw on results that establish the limit behavior of predicates written in the *first* and *second order* logic. Our central viewpoint is that dynamic, global computing systems are not amenable to a "static", completely formal definition of trust. We, rather, believe that trust should be a *statistical*, *asymptotic* concept to be studied in the limit as the system's components grow according to some growth rate. Thus, our main goal is to define trust as an emerging system property that "appears" when a set of properties hold, asymptotically, almost certainly in random communication structures that model computing systems and the interaction between constituent devices. This requires, first, that one adopts a random graph model that best suits the target dynamic system (network). Then a number of properties that model facets of trust are stated using first order logic or some second order logic fragment. Moreover, conditions are established under which these properties appear (or do not appear) in the limit, as the system grows.

The remainder of the chapter is organized as follows. The next section contains the basic random graph models that are currently used to model networks. Section 3 presents properties that model facets of trust using first order logic or some second order logic fragment of graphs. Section 4 contains a formal description of a generic trust model based on the *Intersection Random Graph* model and *fixed radius graph* model. Section 5 presents the conditions under which these models exhibit threshold behavior. Section 6 defines natural properties of these models that emerge though local trust interactions (trust edges of the model). Section 7 uses undecidable statements on random graphs that show the limitations of the formal approach. In Section 8, we discuss an important weakness that arises in any formal trust framework. We conclude in Section 9 with a summary of our discussion and some ideas for future research. Finally, preliminary portions of this work appeared in (Liagkou et al., 2007) and (Liagkou et al., 2009).

2. Random Graph Models For The Global Computing Paradigm

As we discussed above, the departure point of our work is that dynamic, "boundary-transcending"computing systems, are not amenable to a static viewpoint of the trust concept, no matter how this concept is formalized. Thus, our main goal is to define trust as an emerging relationship among entities of the system, that "appears" when a set of properties hold, asymptotically, almost certainly in random communication structures that model computing systems and the interaction between constituent devices. And one of the most well studied and most intuitively appealing formalism for studying *emergent properties* is the *graph*. This trust metric model can be used to evaluate trust assertions in a distributed information system. Generally, directed graphs can be used to represent and answer the following questions: A trusts B, A trusts C, B trusts D, C trusts D, when trust is assumed to be a binary, directed relationship. In order to evaluate trust between two or more entities, we can assign weights (or believe estimates) to the degree of trust given on the trust relationship. The trust as a numerical value, weighted edges can be introduced in the Strust graph model T. These weights can provide primary data for acquiring a trust value. As long as trust values are just complete definable (e.g. A trusts B and C, no trust statement is expressed to all the other entities),

it is quite easy to represent a trust metric in a weighted directed graph and make suitable deductions using, for instance, belief propagation techniques or Bayesian reasoning.
However, things may get complicated if very large community graphs are involved, that evolve in an unpredictable way, such as the WWW society (see (Bollobás, 2001) for a thorough treatment of threshold phenomena in relation to random graph properties).
In this subsection we will refer to the basic random graph models that are currently used to model entities and relations among them as graphs: nodes represent entities and edges among entities represent relations (e.g. a "trust" declaration). But why "random"? Randomness in the graph model has been studied extensively and many rigorous results exist for proving that evolving graphs have a number of interesting, emerging, global properties. But this is a matter of convenience in proving things about big structures, such as the dynamic networks and its trust relationships. Actually, randomness is a way to model the unpredictability of how the network structure changes by the addition (and deletion) of huge numbers of links (communication links or trust relationships in our case) on a daily basis. Since unpredictability without any previous knowledge about possible biases permits the "full randomness" assumption, random graphs may uncover many interesting properties of the network graph.
We will assume from now on, for simplicity, that trust relationships are symmetric and no weights (i.e. trust strength estimates) exist for these relationships. The basic definitions can be extended but we will refrain from doing so in order to exemplify the basic techniques. In what follows, by n we will denote the number of network nodes and by Ω the set of all possible $\binom{n}{2}$ edges between these nodes.

- Model $\mathcal{G}_{n,m}$: select the m edges of G by selecting them uniformly at random, independently of one another from Ω.
- Model $\mathcal{G}_{n,p}$: include each edge of Ω in G independently of the others and with probability p.
- Model $\mathcal{G}_{n,R_0,d}$: generate n points in some d-dimensional metric space uniformly at random and draw an edge between two points only if their distance is at most R_0.
- Model $G_{k,m,p}$: each node i of the k available creates a set S_i by selecting uniformly at random each of the available m objects with probability p. Then an edge is formed between two nodes i, j only if $S_i \cap S_j \neq \emptyset$. This is the random intersection graph model.

There is also another very useful graph model, called the *scale-free graph model* (see (D. Alderson and et al., 2006) for definitions and results related to this model) which is found to accurately model real, fixed topology networks. This model, however, cannot model dynamic, structureless networks and we will not refer to it further in this chapter. Our focus will be the random intersection graph model (see Subsection 5.2).

3. A Brief Introduction To The First And Second Order Language Of Graphs

3.1 First order language of graphs

We are interested in discovering conditions under which a random graph model displays threshold behavior for certain properties that can also be relevant to trust or security issues. In this subsection we will be focused on properties expressible in the *first order language* of graphs. This language can be used to describe some useful (and naturally occurring in applications) properties of random graphs under a certain random graph model using elements of the first order logic.
The alphabet of the first order language of graphs consists of the following (see, e.g., (Spencer, 2001)):

- Infinite number of variable symbols, e.g. $z, w, y \ldots$ which represent graph vertices.
- The binary relations "==" (equality between graph vertices) and "$\sim$" (adjacency of graph vertices) which can relate only variable symbols, e.g. "$x \sim y$" means that the graph vertices represented by the variable symbols x, y are adjacent.
- Universal, $\exists$, and existential, $\forall$, quantifiers (applied only to *singletons* of variable symbols).
- The Boolean connectives used in propositional logic, i.e. $\vee, \wedge, \neg, \Longrightarrow$.

An example of graph property expressible in the first order language of graphs is the existence of a triangle:

$$\exists x \exists y \exists w (x \sim y) \wedge (y \sim w) \wedge (w \sim x).$$

Another property is that the diameter of the graph is at most 2 (can be easily written for any fixed value k instead of 2):

$$\forall x \forall y [x = y \vee x \sim y \vee \exists w (x \sim w \wedge w \sim y)].$$

However, other equally important graph properties, like connectivity, cannot be expressed in this language.
We will now define the important *extension statement* in natural language, although it clearly can be written using the first order language of graphs (see (Spencer, 2001) for the details):

Definition 3.1 (Extension statement $A_{s,t}$). *The extension statement $A_{s,t}$, for given values of s, t, states that for all distinct $x_1, x_2, \ldots, x_s$ and $y_1, y_2, \ldots, y_t$ there exists distinct z adjacent to all x_is but no y_j.*

The importance of the extension statement $A_{r,s}$ lies in the following Theorem. When applied to the first order language of graphs,

Theorem 3.1. *Let G to be a random graph with n nodes and $A_{r,s}$ to be an extension statement, then if $A_{r,s}$ for* all r, s $\lim_{n\to\infty} \mathbf{Pr}[G \text{ has } A_{r,s}] = 1$, *then for every statement A written in the first order language of graphs either* $\lim_{n\to\infty} \mathbf{Pr}[G \text{ has } A] = 0$ *or* $\lim_{n\to\infty} \mathbf{Pr}[G \text{ has } A] = 1$.

The connection between threshold properties and first order logic was first noted by Fagin in the seminal paper (Fagin, 1976).
In Section 4 we will describe a simple trust model based on the intersection random graph model and in Subsection 5.2 we will provide conditions under which this model displays threshold behavior and, thus, has (or has not) certain properties related to trust.

3.2 Second order language of graphs

Although the extension property can be used in order to settle the existence of thresholds for all properties expressible in the first order language of graphs in any random graph model, things change dramatically when properties are considered that are expressed in the *second* order language of graphs.
The second order language of graphs is defined exactly as the first order language (see Section 3.1) except that it allows quantification over subsets of graph vertices (predicates) instead of single vertices. An example of such a property follows (see, e.g., (Jukna, 2001)).

Definition 3.2 (Separator). *Let $\mathcal{F} = \{F_1, F_2, \ldots, F_m\}$ be a family of subsets of some set X. A separator for $\mathcal{F}$ is a pair (S, T) of disjoint subsets of X such that each member of $\mathcal{F}$ is disjoint from either S or from T. The size of the separator is* $\min(|S|, |T|)$.

In the context of trust, this property may be interpreted as follows. Let us assume that $|F_i| = 2$, modeling an edge of a graph. Thus, the sets F_i model a graph's links between pairs of nodes. With this constraint, the separator property says that in a graph there exist two disjoint sets of nodes S and T such that any set of two adjacent (i.e. communicating) nodes is disjoint from either S or T. In other words, it is not possible to have one node belonging to one of the two disjoint sets S and T and the other node belonging to the other. This might mean that no two communicating nodes are authenticated by two different authentication bodies (the two disjoint sets of nodes). Thus, the two nodes can trust each other more since they are not authenticated by two disjoint (i.e. unrelated) authentication bodies. Each of the two disjoint sets may form, for instance, Certification Authority (CA) providing authentication services.
In order to cast the separator property into the language of graphs, we set X to be a set of vertices and the subsets F_i to be of cardinality 2 so as to represent graph edges. Then the separator property can be written in the framework of the second order language of graphs as follows

$$\exists S \exists T \forall x \forall y [\neg(Sx \wedge Tx) \wedge (Axy \rightarrow \neg(Sx \wedge Ty \vee Sy \wedge Tx)]. \tag{1}$$

Let us define another property:

Definition 3.3 (Trusted representatives)**.** *A graph G has the trusted representatives property if there exists a set of vertices such that any vertex in the graph is an adjacent with at least one of these vertices.*

A formal definition using second order logic is the following

$$\exists S \forall x \exists y [Axy \wedge Sy]. \tag{2}$$

The extension statement, cannot, unfortunately, be used in order to examine whether (and under which conditions on the random graph model parameters) the separator property or the trusted representatives property is a threshold property since these properties cannot be written in the first order language of graphs.
However, in 1987 Kolaitis and Vardi initiated in (Kolaitis & Vardi, 1987) a research project in order to characterize fragments of the second order logic that display threshold behavior (i.e. they have a 0-1 law). The interested reader may consult the review paper (Kolaitis & Vardi, 2000) by the same authors. Without delving into the details, one of the important conclusions reached at by this project is that there are second order fragments that do not have a threshold behavior while other second order fragments do.
Let Σ_1^1 denote the existential second order logic (i.e. formulas contain only existential quantification over second order variables, that is sets). Let FO denote the first order logic formalism and $\mathcal{L}$ be any fragment of FO. Then a $\Sigma_1^1(\mathcal{L})$ sentence over a vocabulary $\mathcal{R}$ is an expression of the form $\exists \mathcal{S} \phi(\mathcal{R}, \mathcal{S})$, where $\mathcal{S}$ is a set of relation variables and $\phi(\mathcal{R}, \mathcal{S})$ is a first order sentence on vocabulary $(\mathcal{R}, \mathcal{S})$. In general threshold behavior is not displayed by Σ_1^1 (see (Kolaitis & Vardi, 2000)). Thus, in order to discover fragments of Σ_1^1 that do have such a behavior, a restriction is imposed on the first order part (i.e. the sentence ϕ written in $\mathcal{L}$) of the sentences considered. This restriction refers to the pattern of quantifiers that appear in the first order sentence ϕ. Some restricted first order logics that have been studied in connection to Σ_1^1 are the following:

1. The *Bernays-Schönfinkel class*, which is the set of all first order sentences with quantifier prefixes of the form $\exists^* \forall^*$ (that is, the existential quantifiers precede the universal quantifiers).

2. The *Ackermann class*, which is defined as the collection of first order sentences of the form $\exists^*\forall\exists^*$ (that is the quantification prefix contains only one universal quantifier.
3. The *Gödel class*, which is defined as the collection of first order sentences of the form $\exists^*\forall\forall\exists^*$ (that is, the prefix contains two consecutive universal quantifiers).

The separator property defined by (1) belongs to the second order fragment Σ_1^1(Gödel) since it contains (in the first order part) two consecutive universal quantifiers. On the other hand, the trusted representatives property defined by (2) belongs to the second order fragment $\Sigma_1^1(Ackermann)$ since it contains a single universal quantifier.
The trusted representatives property can be proved to be a threshold property since the second order logic fragment $\Sigma_1^1(Ackermann)$ has a threshold behavior in general (see (Kolaitis & Vardi, 2000)). This means that, asymptotically, it holds with either probability 0 or 1 depending on the random graph model parameters. On the other hand, the separator property is not guaranteed to be a threshold property since the Σ_1^1(Gödel) second order logic fragment does not display a threshold behavior in general (see (Kolaitis & Vardi, 2000)).
Thus, sentences (properties) that can be written in fragments of second order logic that have a threshold behavior (e.g. $\Sigma_1^1(Ackermann)$) are threshold properties. However, some second order logic fragments allow the construction of sentences that have no limiting probability and, thus, are not threshold properties.

4. A Generic Trust Model Based On Threshold Laws For Mathematical Logic

As we mentioned in the Introduction, trust is a difficult concept to formalize and handle. What is more, our target framework of global/dynamic computation clusters does not seem to allow a static view of the trust concept, regardless of the way in which this concept is formalized. Our viewpoint is that trust should be a statistical, asymptotic concept to be studied in the limit, as the system's components grow according to some growth rate.
The random graph models described in Section 2, each with its own definition of node adjacency, seem to be suitable candidates for studying the trust concept as the asymptotic appearance of specific trust patterns in the graph. Thus, our practical viewpoint of trust in a dynamic, global computing system is the following (see, also, Figure 1):

- First one adopts a suitable random graph model that best suits the target dynamic system (network). For instance, if graph nodes model system components (e.g. sensors) that move about in Euclidean space and adjacency between pairs of them is decided according the their transmission range, the fixed radius model is a good choice for modeling the network (see, e.g., (Liagkou et al., 2006)). If, however, one is interested in patterns arising in the Internet graph, the preferential random graph model is best.
- Secondly, one is focused on defining a number of properties that model facets of trust using first order logic or some second order logic fragment. Examples of such properties is the triangle property given in Section 3.1 and the separator and trusted representatives properties defined in 1 and 2 in Section 3.2. If the property can be cast into the first order language of graphs, then one is certain that this is a certain property that either is possessed almost certainly by the growing system or it is not possessed almost certainly, depending on its monotonicity. Then the interesting part is to establish relationships among the random graph model parameters that allow the almost certain appearance or disappearance of the property for random systems generated according to the chosen random graph model (this will be undertaken for the intersection graph

model in Section 5.2). If the property does not seem to be amenable to definition within the realm of the first order logic, then proceeds to the next step.

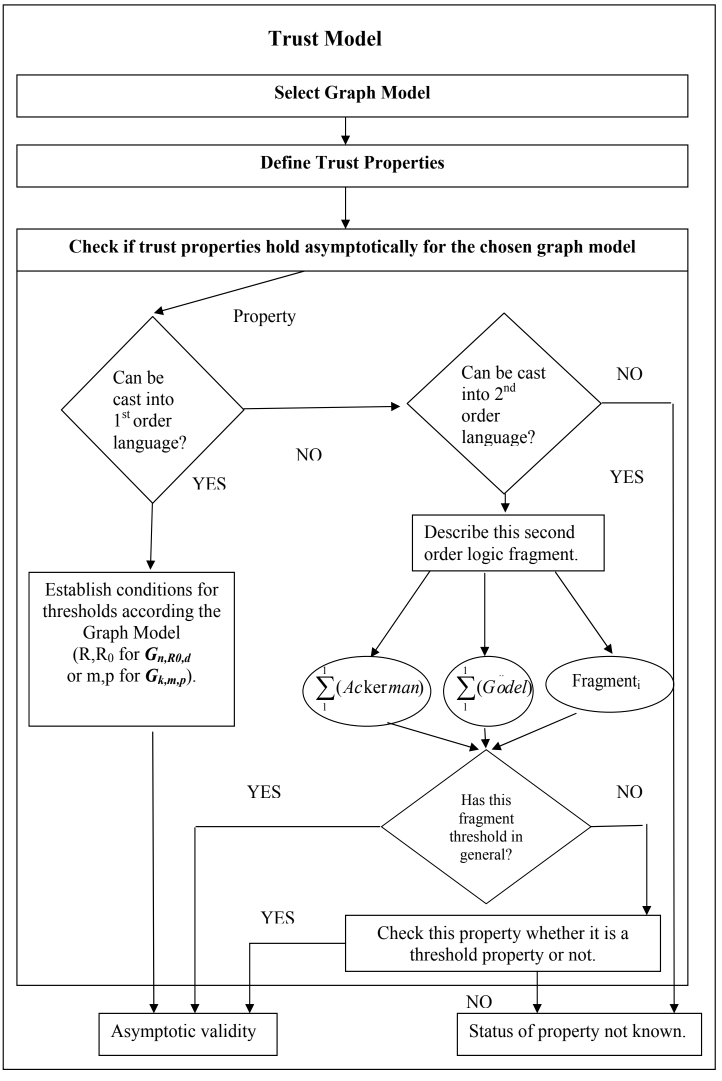

Fig. 1. General Logic-based trust approach

- Following the second step, if the property under consideration can only be written using second order logic, then one examines whether the property can be cast into the language of a fragment of the second order logic that has a threshold behavior (e.g. the $\Sigma_1^1(Ackermann)$ class). Then one is certain that as the system grows the property holds asymptotically almost certainly or almost never (again depending on its monotonocity). However, if the property seems to be describable only in a second order logic fragment that, in general, does not have a threshold behavior (e.g. Σ_1^1(Gödel)) then this property should be further examined as to whether it is a threshold property or not. Such a property, called *Kernel* (see below for a definition) is given in (Bars, 1998) for the $\mathcal{G}_{n,p}$ model with fixed p. It is interesting to define second order properties related to trust for a random graph model that have no threshold behavior since they are guaranteed to hold for a positive fraction of the random structures allowed by a random graph model.

The *Kernel* property, which we believe can be the prototype for discovering other non-threshold properties, is defined in the context of directed graphs. The language of directed graphs is the same as the language of undirected graphs with only difference that the predicate $A_{x,y}$ that signifies adjacency between x and y is not symmetric. A random digraph, according to model $\mathcal{G}_{n,p}$ is constructed by having each of the possible, directed edges being chosen for inclusion independently of each other, with constant probability p. Then a kernel in the produced directed graph is a subset U of the set of vertices such that no edge exists between vertices within U while for each vertex outside U there exists an edge from this vertex to some vertex within U. This property is given below, written in the second order language of graphs (see (Bars, 1998)):

$$\exists U[(\forall x \forall y((Ux \wedge Uy) \rightarrow \neg A_{x,y})) \wedge (\forall x \exists y(\neg Ux \rightarrow (Uy \wedge A_{x,y})))]. \qquad (3)$$

The property in (3) is written in $\Sigma_1^1(FO^2)$, with FO^2 being the fragment of first order logic allowing propositions containing at most 2 variables.

5. Threshold Behavior

In this section we study the threshold behavior of these two models, in order to define the first order properties related to trust and to specify the conditions for ensuring properties' validity or non-validity. Let us firstly describe the threshold phenomena in relation to random graph properties

5.1 The delicate balance between validity and non-validity of statements about large relationship structures: threshold phenomena

The concept of a threshold function or transition point in connection with properties of combinatorial objects, such as graphs, is well understood in discrete mathematics and combinatorics (see (Bollobás, 2001) for a thorough treatment of threshold phenomena in relation to random graph properties). However, the suggestion to look at this concept from a fresh perspective was given by Cheeseman, Kanefsky, and Taylor in (Cheeseman et al., 1991). One of the problems they examined was a problem equivalent to 3-SAT, in the complexity theoretic framework of NP-completeness, i.e. they are both computationally intractable and if one of them could be solved efficiently, then a multitude of other problems believed to be computationally intractable would also be solvable efficiently. The problem was that of colouring the vertices of a graph with three colours, also known as 3-COLOURING, in a way such that no two adjacent vertices are assigned the same colour. The graphs that can be coloured with 3 colours are called 3-colourable. Note that in our context, 3-colorability of a graph is a global property that is composed of the conjunction of a number of many local relationships (graph edges) and, thus, it may views as a global emerging property that arises when several local conditions hold simultaneously.

In the theory of random graphs (see (Bollobás, 2001)), we are interested in whether a randomly formed graph possesses a property, such as being 3-colourable, or not. A random graph with n vertices and m edges is most commonly formed according to the following model: from the set of possible $\binom{n}{2}$ edges, select uniformly and without replacement m edges to belong to the graph. Now a natural question that arises is the following: for various values of m (chosen edges), what is the probability that a random graph of m edges possesses the property in question as n tends to infinity? Let us consider the property of a graph being colourable with 3 colours. If $m = \omega(n)$, meaning that as n grows, m grows so that the ratio m/n tends to infinity, it can be easily proved using the first moment method that was applied above to the 3-SAT problem that with probability tending to 1, a random graph with m edges will not be 3-colourable. On the other hand, if $m = o(n)$, meaning that as n tends to infinity m/n tends to 0, we can use a result from the theory of random graphs (see the book of Bollobás (Bollobás, 2001), Corollary 5.8, page 105) that states that in this case, with probability tending to 1, every component of a random graph with m edges is either a tree or a unicyclic graph (i.e. a non-chordal ring with trees attached to some of its nodes). But this means that the graph *can* be coloured with at most three colours.

From the above discussion, we conclude that the function $f(n) = n$ marks a, so-called, *threshold* area, in the sense that if the number of edges in a randomly formed graph grows slower or faster than $f(n)$ then we observe in each case a different behaviour, with probability that tends to 1. We, then, say that $f(n) = n$ is a *threshold function* for the property of 3-colourability. What happens, however, when $m = \Theta(n)$, i.e. when m/n tends to a positive constant value r? Well, in this case it may or may not be true that, almost certainly, a randomly formed graph with m edges can be coloured with 3 colours. The key factor is the exact value of r, the constant itself. Therefore, we shift our attention to the study of the *"micro-threshold"* behaviour, i.e. we fix the order of growth of m to be n and we focus on discovering the ranges of r that correspond to graphs that are or are not 3-colourable with probability that tends to 1.

To return to Cheeseman, Kanefsky, and Taylor, their experiments demonstrated that for values of r outside the interval $(4, 6)$ (approximately), the random graphs with rn edges were either almost all 3-colourable or almost none 3-colourable. This suggests that there may by some value for r in $(4, 6)$ for which we may observe an abrupt transition from almost certain 3-colourability to almost certain non 3-clourability of the random graphs with rn edges. Indeed, their experiments indicated that the transition takes place around the value $r = 5.4$.

The morale from this is that an assertion about a given system, e.g. that a graph is 3-colorable, may change dramatically with only a small linear change in the number of relations (graph edges). This sensitivity precludes the extraction of safe conclusions with regard to the property within a unpredictably evolving and growing environment. Small local additions of relations, may result in global changes in the global state into one or the other direction and one should be cautious of this abrupt change of states. In addition, algorithmic complexity dictates that stating whether the one or the other state prevails, is algorithmically intractable in a number of interesting combinatorial structures, like the graphs we studied in this section.

5.2 Threshold behavior of the intersection graph model and trust

Here we study the threshold behavior of the intersection graph model with regard to properties expressible in the first order language of graphs.
We will assume that for the edge probability p it holds $p \neq 0, 1$ since in this case the extension property cannot hold for any random graph model.
This model was presented in (Nikoletseas et al., 2004) with the name *General Random Intersection graph model*. The edges between vertices in this model are formed as follows. Let as consider a universe $M = \{1, 2, \dots, m\}$ of elements and a set of vertices $V = \{1, 2, \dots, k\}$. Then if we assign independently to each vertex $j \in V$, a subset S_j of M by choosing each element $i \in M$ independently with probability p_i and insert an edge between two vertices j_1, j_2 if and only if $S_{j_1} \cap S_{j_2} \neq \emptyset$ then the resulting graph is an instance of the general random intersection graph $\mathcal{G}_{k,m,p}$, with p_i. In our work, we set all p_is to have the same value, p, abusing the notation. Obviously, the probability of having an edge between two vertices is equal to $1 - (1 - p^2)^m$.

Lemma 5.1. *The probability that $A_{s,t}$ fails for a random graph of the $\mathcal{G}_{k,m,p}$ model is bounded from above as follows:*

$$\mathbf{Pr}[A_{s,t} \textit{ fails in } \mathcal{G}_{k,m,p}] \leq \binom{k}{s+t} \left[1 - P_e^s (1 - P_e)^t\right]^{k-(s+t)} \tag{4}$$

with $P_e = 1 - (1 - p^2)^m$.

Theorem 5.2. *For the random model $\mathcal{G}_{k,m,p}$, with m, p functions of k, three sufficient condition for the right-hand side of (4) to tend to 0 are the following:*

- $\lim_{k\to\infty} p^2 m = \textit{constant} \neq 0$.
- $\lim_{k\to\infty} p^2 m = 0$ *and* $p^2 m \gg \frac{1}{\ln(k)}$.
- $\lim_{k\to\infty} p^2 m = \infty$ *and* $p^2 m \ll \ln(k)$.

Proof. From Inequality (4), it follows that

$$\mathbf{Pr}[A_{s,t} \text{ fails in } \mathcal{G}_{k,m}] \leq \binom{k}{s+t} \cdot \left(\exp\left[-(1-(1-p^2)^m)^s[(1-p^2)^m)]^t[k-(s+t)]\right]\right). \tag{5}$$

We will establish conditions on the parameters k, m, p that suffice to force the right-hand side of (9) to tend to 0. These conditions will define ranges on k, m, p that suffice in order to ensure that the intersection random graph model displays threshold behavior.
In order to have the right-hand side of (9) to tend to 0, for any fixed s and t, it suffices to ensure that

$$(1-(1-p^2)^m)^s[(1-p^2)^m)]^t[k-(s+t)] \to_{k\to\infty} \infty. \tag{6}$$

Case 1 Assume, first, that $\lim_{k\to\infty}(1-p^2)^m$ is a constant c, $0 < c < 1$. This happens only if $p^2 m$ is (or tends to) a constant different from 0. In this case, Condition (6) holds since the expression there is $\Theta(k)$.

Case 2 Assume, now, that $\lim_{k\to\infty}(1-p^2)^m = 1$, which holds only if p^2m tends to 0. In this case we can apply the approximation $(1-p^2)^m \sim 1-p^2m$. Then the expression in (6) is, asymptotically, equal to $k(p^2m)^s$. Thus, a sufficient condition for (6) to hold is to have $p^2m \gg \frac{1}{\ln(k)}$.

Case 3 Finally, assume that $\lim_{k\to\infty}(1-p^2)^m = 0$, which occurs if p^2m tends to infinity. Then for Condition (6) to hold it suffices to ensure that

$$k(1-p^2)^m$$

converges to 0. Equivalently, we need to ensure that

$$(1-p^2)^m \gg \frac{1}{k}.$$

Taking logarithms, we need to have

$$m\ln(1-p^2) \gg -\ln(k). \tag{7}$$

Since p tends to 0, we can approximate $\ln(1-p^2)$ with $-p^2$. Thus, (7) becomes

$$m(-p^2) \gg -\ln(k)$$

which holds if $mp^2 \ll \ln(k)$ completing the proof of the theorem. □

□

5.3 Threshold behavior of the fixed radius random graph model

In (Liagkou et al., 2006) we proved that fixed radius random graph model has a threshold behavior in order to introduce a key distribution scheme. Here we use the following Lemma and Theorems in order to demonstrate that a number of global system properties related to trust can be described in the first order language of graphs and that they hold with probability 1 for certain ranges of the fixed radius random graph model.

Lemma 5.3. *For the 2-dimensional sphere (circle) the probability that $A_{s,t}$ fails for $\mathcal{G}_{n,R_0,d}$ is bounded from above as follows:*

$$\mathbf{Pr}[A_{s,t} \textit{ fails in } \mathcal{G}_{n,R_0,2}] \le \binom{n}{s+t}\left[1-D_2(R_0)^s(1-D_2(R_0))^t\right]^{n-(s+t)}. \tag{8}$$

Theorem 5.4. *If $\sigma = \frac{R_0}{2R} = c$ is a constant, $0 < c < 1$, then Equation (8) tends to 0. If $\sigma = \frac{R_0}{2R} = f(n) = \omega(\frac{1}{\sqrt{n}})$, then Equation (8) also tends to 0.*

Proof. From Equation (8), it follows that

$$\mathbf{Pr}[A_{s,t} \text{ fails in } \mathcal{G}_{n,R_0,2}] \le \binom{n}{s+t} \exp\left[-D_2(R_0)^s(1-D_2(R_0))^t(n-(s+t))\right]. \tag{9}$$

Our goal is to find a condition on c such that the right-hand side of (9) tends to 0. Then $\mathbf{Pr}[A_{s,t}$ fails in $G(n,R,2)]$ tends to 0 and, consequently, $\mathbf{Pr}[A_{s,t}$ holds in $\mathcal{G}_{n,R_0,2}]$ tends to 1 establishing the fact that any first order property holds, asymptotically, in $\mathcal{G}_{n,R_0,2}$ with probability 1 or 0.

Case 1 Let σ be a constant c, $0 < c < 1$. Then $D_2(R_0)$ is a constant too. Thus, the exponential factor of the right-hand side of Equation (9)

$$\exp\left[-D_2(R_0)^s(1-D_2(R_0))^t(n-(s+t))\right] \tag{10}$$

tends to 0, for fixed s,t and n tending to infinity. Therefore, the probability $\mathbf{Pr}[A_{s,t}$ fails in $\mathcal{G}_{n,R_0,2}]$ also tends to 0.

Case 2 Let, now, $\sigma = f(n) < 1$, a function of n tending to 0. Then using power series analysis around 0, we obtain from (9) the following:

$$\begin{aligned} D_2(R_0) &= 4\sigma^2 + \frac{1}{2\pi}\sigma\,(4-4\sigma)^{\frac{3}{2}} \\ &- \frac{3}{\pi}\sigma\sqrt{4-4\sigma^2} + \frac{2}{\pi}\arcsin\sigma - \frac{8}{\pi}\sigma^2\arcsin\sigma \\ &= 4\sigma^2 \\ &- \frac{32}{3\pi}\sigma^3 + \frac{16}{15\pi}\sigma^5 + O(\sigma^6). \end{aligned} \tag{11}$$

The term $D_2(R_0)^s(1-D_2(R_0))^t$ in the exponent in (10) can be approximated as follows:

$$\begin{aligned} D_2(R_0)^s(1-D_2(R_0))^t &= 4\,s\,\sigma^2 - \frac{32\,s}{3\,\pi}\sigma^3 \\ &- [16\,s\,t + 8\,s\,(s-1)]\,\sigma^4 \\ &+ \left[\frac{256\,s\,t}{3\,\pi} + \frac{16\,s}{15\,\pi}\right] \\ &+ \left[\frac{128\,s\,(s-1)}{3\,\pi}\right]\sigma^5 \\ &+ \mathrm{O}(\sigma^6) \end{aligned} \tag{12}$$

with s,t constants. Then, from (10) and (12), it follows that if $\sigma = f(n) = \omega(\frac{1}{\sqrt{n}})$, then (10) tends to 0, for any s,t, completing the proof. □

□

The generalization, now, follows readily:

Theorem 5.5. *Let $\sigma = \frac{R_0}{2R} = c$ be a constant, $0 < c < 1$. Then for any first order property A, then $\mathbf{Pr}[\mathcal{G}_{n,R_0,d}$ has $A]$ tends to 1 or 0. If $\sigma = \frac{R_0}{2R} = f(n) = \omega(\frac{1}{\sqrt[d]{n}})$, then $\mathbf{Pr}[\mathcal{G}_{n,R_0,d}$ has $A]$ tends to 1 or 0 too.*

Although the property of forming a connected graph cannot be described in the first order theory of graphs, in (Gupta & Kumar, 1998) it is shown that for slighter larger values of σ, the network is almost certainly connected. More specifically, we only need to increase the threshold probability (in the 2-dimensional case) from $\frac{1}{\sqrt{n}}$ to $\frac{\sqrt{\log(n)}}{\sqrt{n}}$ to, also, ascertain connectivity in the resulting graph.

6. Trust Properties

We will now propose a number of trust-related properties that can be studied in the context of the random intersection graph model and the fixed radius random graph model. We can discover these trust properties along two directions using the ideas proposed in the previous sections. The first direction consists in discovering a number of first order properties related to trust, that emerge through the local trust interactions (trust connections of the models), and define ranges of the model parameters that lead to the almost certain asymptotic validity or non validity of the global property of interest.

6.1 Trust properties of intersection graph model

Let us assume that we have a $\mathcal{G}_{k,m,p}$ random graph, interpreting its parameters in the following way. We have k available computing agents and m resources (e.g. service access points or computer ports, located in some server). According to the model, each of the k agents selects uniformly at random from within the set of the m resources, each of which selected independently of the others with probability p. Then two agents are connected with a "trust" edge whenever their selections contain at least one shared service (i.e. two agents do not trust directly each other - they trust each other only if they use at least one common resource). Note that the set of services could even be a set of trusted third parties that can certify the identity of each agent. Then two agents trust each other if they "use" at least one trusted resource (the trust relationship is symmetric, although in general this is not necessarily true).
From this point, we can proceed along two directions using the ideas proposed in the previous sections.
The first direction consists in discovering a number of global system properties related to trust, that emerge through the local trust interactions (trust edges of the model), and define ranges of the model parameters that lead to the almost certain asymptotic validity or non validity of the global property of interest.
For concreteness, let us define the following first order property:

$$\forall x \exists y [A_{x,y}] \tag{13}$$

which states that for each node x there exists at least one other node such that the two nodes trust each other. Since this property is monotone increasing, if the model parameters k, m, p obey the conditions of Theorem 5.4 then as the node population increases, the property stated above holds with probability tending to 1.
Another property that can be defined is the following:

$$\forall x \forall y \forall z [A_{x,y} \wedge A_{y,z} \rightarrow Axz] \tag{14}$$

which states that the trust relationship is transitive. Again, if the conditions on the random intersection graph model parameters hold, then in the limit the trust relationship is transitive with probability tending to 1. Similarly, the trusted representatives property holds for the random intersection graph model (see discussion in Section 3.2).

6.2 Trust properties of fixed radius random graph model

Suppose that we have n agents randomly distributed within a circle of radius R_0 . We first define a circle of radius C cantered at each agent. Our fixed radius random graph with n agents is formed so as to include "trust" edges between agents only if their distance is at most 2C Thus two agents establish a trusted connection if their cycles (of radius C) are intersected.

Let us now define some first order properties related to trust using the threshold properties of the fixed radius graph model. In this context, $R_0 = 2C$, that is two agents that trust each other if their ranges intersect, which occurs if their distance is at most $2C$. Thus according Theorem 5.4, $\sigma = \frac{C}{R}$. Let $C = C(n)$ and $R = R(n)$ be functions of n tending to infinity and set $\sigma(n) = \frac{C(n)}{R(n)} = o(1)$. The assumption of $R(n)$ and $C(n)$ tending to infinity reflects the fact that we have a large scale distributed system. The assumption $\sigma(n) = o(1)$, however, reflects the fact that we should allow the agents to trust only the agents that distribute within R(n) area. Consider, now, possible ranges for $\sigma(n)$. According to Theorem 5.4, if $[\sigma] = \omega(\frac{1}{\sqrt{n}})$ then the extension property holds with probability approaching 1 as the number of agents increases. This means that all properties expressible in the first order language of graphs hold (asymptotically with n) either with probability 1 or 0. Especially, properties that are *monotonically increasing* (i.e. the probability of the property holding increases with increasing $\sigma(n)$) hold with probability 1 while their complementary properties hold with probability 0. What we need to do next is to define *trust* properties, which can be expressed in this graph language.

Let us consider the following property: *every two vertices have a common trust agent*. If this property holds, then for *each* pair of agents that establish a trust connection there exists another trusted identity . This may cause problems since it increases the number of trusted parties without reason. As they both trust a third agent it is better one of them an indirectly trust connection with the third one. . Setting $\sigma(n) = \frac{C(n)}{R(n)} = O(\frac{1}{\sqrt{n}})$, and since this property is monotone increasing, it holds with probability tending to 0. Thus its complementary property, which is a trust' property, holds with probability 1.

The second direction along which one can proceed is, in some sense, the opposite of the direction outlined above. The goal is not to establish conditions for ensuring almost certain validity or non-validity of some first order property related to trust but, on the contrary, to state higher order properties in the second order language of graphs (like the separator or trusted representatives property given in Section 3.2) and show that the properties have no limiting probability, i.e. they cannot be threshold properties. Such a property, being not a threshold property, leads a complex system to some kind of equilibrium, as the system grows. In both directions given above, the central idea is that trust is global property characterized by local interaction between system entities.

7. Probability Theory - Undecidable Probabilities

Theorem 7.1 (Trachtenbrot-Vaught Theorem (Trachtenbrot, 1950))**.** *There is no decision procedure that separates those first order statements S that hold for some finite graph from those S that hold for no finite graph.*

With regard to random graphs now which, as we show, in conjunction with the first and second order language of graphs, can be used to express, formally, complex relationships that can be related to trust, we have the following result (see (Dolan, 1992)):

Theorem 7.2. *There is no decision procedure that separates those first order statements S that hold almost always for the random graph $\mathcal{G}_{n,p}$ from those for which $\neg S$ holds almost always.*

This theorem is targeted to $\mathcal{G}_{n,p}$ random graphs, with $p = n^{\alpha}$, α being a rational number between 0 and 1. In summary, for any first order statement A about a finite graph, a first order statement A^* is given that holds almost always in $\mathcal{G}_{n,p}$, *if* A holds for some finite graph, while it never holds, if A holds for *no* finite graph. Now, if a formal procedure (algorithm) existed for deciding such statements for the $\mathcal{G}_{n,p}$ model, then relationship between A and A^* would allow using the procedure to separate those first order statements A that hold for some finite graph from the statements that hold for no finite graph, contradicting the Trachtenbrot-Vaught Theorem.

More specifically, let us consider the following statement S: There is no isolated vertex in the graph, which can be written as $\forall y \exists z (y \sim z)$. Let S^* be the corresponding statement, for the random graph $\mathcal{G}_{n,p}$ with $p = n^{-2/5}$ (see (Dolan, 1992)):

$$\exists x_1 \exists x_2 \exists x_3 \exists x_4 [\forall y \mathrm{MEM}(y; x_1, x_2, x_3, x_4) \Longrightarrow \exists z \mathrm{MEM}(z; x_1, x_2, x_3, x_4) \wedge \mathrm{ADJ}(y, z)]$$

with MEM and ADJ the following first order language predicates:

$$\mathrm{MEM}(y; x_1, x_2, x_3, x_4) \Longleftrightarrow \exists z[(z \sim x_1) \wedge (z \sim x_2) \wedge (z \sim x_3) \wedge (z \sim x_4) \wedge (z \sim y)]$$

$$\mathrm{ADJ}(u, v) \Longleftrightarrow \mathrm{MEM}(u; x_1, x_2, x_3, x_4) \wedge \mathrm{MEM}(v; x_1, x_2, x_3, x_4) \wedge \exists t \mathrm{MEM}(t; x_1, x_2, u, v).$$

$$\lim_{n \to \infty} \mathbf{Pr}[G_{n,p} \text{ has } S^*] = \begin{cases} 0 \text{ if } S \text{ holds for no finite graph,} \\ 1 \text{ if } S \text{ holds for some finite graph.} \end{cases} \tag{15}$$

Then a decision procedure that could differentiate between statements that hold almost always in $\mathcal{G}_{n,p}$ and the statements whose negation holds almost always, would provide a decision procedure to differentiate between those statements S that hold for *some* finite graph and those that hold for no finite graph, contradicting the Trachtenbrot-Vaught Theorem.

The morale of this discussion is that it may not even possible to mechanically analyze whether a given state of affairs (e.g. trust assertion) or its negative, within the world of discourse, is expected to almost certainly appear. Thus, it may be the case that one may have to observe the target world for sufficiently much time in order to be able to make a safe prediction about the state of affairs that will finally prevail in the limit.

8. The Self-referential Nature Of Trust

Finally, in this section, we discuss an important weakness that arises in any formalism, when it is sufficiently powerful to be able to "talk about itself", i.e. to contain statements about its expressive and deductive power (i.e. derivable statements).

According to the famous incompleteness theorem of Gödel, any formal system powerful enough to encompass the Peano axioms, contains statements for which neither the statement or its negation can be proved using the axioms and deductive rules of the formal system. In other words, there are truths and valid statements that cannot be asserted, using the formalism and its derivation rules alone. Another expression of this "self-reference" phenomenon, from the point of view of computability theory this time, was given by Alan Turing in 1936 who described a universal computation machine model. In his famous work *On computable numbers, with an application to the Entscheidungsproblem* Turing defined a mathematical model for a device that performs mechanical calculations, later named *Turing machine* after its inventor. This suprisingly minimal, yet maximally powerful, model consisted simply of a infinite tape divided into cells each holding a particular symbol (say 0 or 1), a tape head that can move about the tape reading or writing symbols and, most important, a finite control able to decide on the next thing to do based on the current machine state and the symbol currently under the tape head. The first success of this simple model of algorithmic computation came immediately: Turing proved that no Turing machine and, hence, no algorithm according to *Church's Thesis* exists to decide whether another Turing machine halts when it starts computing with a specified input putting an end to Hilbert's grand program of mechanizing mathematics. The proof, actually, is a computational version of the proof of Gödel, which was cast within the logic calculus formalism. (We would like to urge the interested reader to consult (van Heijenoort, 1967) for an excellent account of the developments that paved the way to the rich theories of Computation and Complexity and (Herken, 1995) for a most comprehensive presentation of Computation and Complexity theory as it stands today.)

We can modify the main argument of the two historic results by Gödel and Turing, so as to give a glimpse of the inherent limitations of formalisms with respect to trust definition and manipulation as follows. We recall, that for our purposes trust is a property, a predicate more precisely, that dictates that the involved entities are in a certain state with regard to each other, i.e. the predicate holds.

Let us assume that we have defined a set of trust axioms that we believe are applicable in the situation at hand. For instance, these axioms may include the fact that in our world of discourse trust has the transitivity property, i.e. from $T(x,y)$ and $T(y,z)$ we may deduce $T(x,z)$. We would like to be able to test whether the trust property holds among some other set of entities, by exploiting the axioms and the deduction mechanisms of our formalism. We may recursively enumerate the possible axioms (given trust assertions) of our world of discourse (assumed to be finite) into strings, $w1, w_2, \ldots$. We may also enumerate the possible deduction mechanisms (algorithms) that start from the axioms, apply a set of derivation rules, and then reach a decision with respect to whether a certain trust assertion among entities of our world of discourse is true or not. Then, using an argument similar to Turing's, we may show that no universal trust derivation process may exist that starts from a description of the world of discourse (axioms plus derivation rules) and decides whether a trust assertion follows or not.

9. Conclusions and Directions For Further Research

Trust has been one of the cornerstones of the success of modern society in building well-organized groups of people working towards their own wealth as well as that of theirs peers. This traditional notion of trust, however, has two basic characteristics: i) it is based on personal contact, and ii) frequently, it cannot be explained.

Today, it is impossible to have personal information about any entity (either human or a machine offering a service) of the huge and ever expanding dynamic computing society, with which we may want to communicate or perform a transaction. Thus, we would like to rely on rules as well as automated deductive procedures as to whether we should trust an entity or not.

In this chapter we have reviewed a number of formalisms with respect to their expressive and deductive power when describing large combinatorial structures, where the structure consists of a number of entities as well as trust assertion among them. Initially we attempted to provide a practical and viable definition of trust for dynamically changing computing environments that can be described within the global computing paradigm. Our view is that trust can be reduced to a number of properties that appear as a limiting behavior in systems under certain conditions. These systems are modeled within the formalism of a random graph model according to the context of the target system. Then the properties can be written formally using the first and second order language of graphs. If the properties can be written in the first order language of graphs then one can use the extension statements in order to establish the conditions under which the model displays threshold behavior and, thus, all the properties hold asymptotically with either probability 0 or 1.

On the other hand (and, perhaps, more interestingly) if a property cannot be written in the first order language of graphs then one may try to see if it can be defined within the vocabulary of a second order logic fragment that has threshold behavior. Otherwise, the question of whether the property holds almost certainly or not remains open and needs the application of a more difficult to apply methodology as the one used for proving that the Kernel property is not a threshold property (see (Bars, 1998)). Our view is that in order to study trust within the realm of dynamically changing complex computing systems one has to resort the discovery of formally definable trust properties (that are apt for the application at hand - e.g. the separator property) and see what happens when the system grows.

Finally, we saw that each of the formalisms has some weaknesses in handling trust in complex, large environments containing a huge number of entities that interact unpredictable (almost randomly). Our position is that these observations seem to hint that reliance on formalism alone is not the answer to the problem of defining and manipulating trust. Rather, entities should better focus on including fast heuristics as well as approximations to reality (even accepting trust in some cases axiomatically, e.g. to avoid the incompleteness pitfalls of powerful formal deductive systems). Moreover, it seems that trust will rely, for some time (until we manage to define it alternatively) on what it relied traditionally for the past few centuries: personal experience, public guidance from organizations and governments, creation of awareness groups, and avoiding trusting an entity whenever one is not totally sure about trusting it (educated decisions). Otherwise, formal trust may either be unattainable (e.g. incompleteness results about formalisms) or hard to verify (NP-completeness results from computational complexity).

One possible research direction could be the design of a kind of reductions among second order properties, the *Kernel* being the archetypal one, that can be used to show that other properties also do not have a threshold behavior (much like NP-completeness results) avoiding the complexity of the proof for the Kernel property. Another possible direction of research is to define random graph models that seem to hinder the appearance of threshold properties written in some second order logic fragment. This would help, for instance, to define non-desirable properties (for trust) and show that they cannot possibly hold with probability 1 as the system grows.

We hope that our work will be a first step towards defining a methodology for studying a variety of properties (not only related to trust) using suitable random graph models and then look at the produced (by the model) systems not individually (which is impossible in a rapidly changing environment) but collectively in the limit.

10. References

Bars, J.-M. L. (1998). Fragments of existential second-order logic without 0-1 laws, *13th IEEE Symp. on Logic in Computer Science*, pp. 525–536.

Blaze, M., Feigenbaum, J. & Lacy, J. (1996). Decentralized trust management, *IEEE Symposium on Security and Privacy*, Oakland (CA, USA), pp. 164–173.

Bollobás, B. (2001). *Random Graphs*, second edition edn, Cambridge University Press.

Cheeseman, P., Kanefsky, B. & Taylor, W. (1991). Where the really hard problems are, *International Joint Conference on Artificial Intelligence*, pp. 331–337.

D. Alderson and, J. D., Li, L., Tanaka, R. & Willinger, W. (2006). Towards a theory of scale-free graphs: Definition, properties, and implications (extended version), *Technical Report CIT-CDS-04-006,*, Engineering & Applied Sciences Division California Institute of Technology, USA.

Dimitrakos, T. (2001). System models, e-risk and e-trust. towards bridging the gap?, *Towards the E-Society: E-Business, E-Commerce, and E-Government*.

Dimitrakos, T. & Bicarregui, J. (2001). Towards a framework for managing trust in e-services., *Fourth International Conference on Electronic Commerce Research*, Vol. 2, pp. 360–381.

Dolan, P. (1992). Undecidable statements and random graphs, *Annals of Mathematics and Artificial Intelligence* **6**: 17–26.

Eschenauer, L., Gligor, V. & Baras, J. (2002). On trust establishment in mobile ad-hoc networks, *Security Protocols Workshop*, Cambridge, UK, pp. 47–66.

Fagin, R. (1976). Probabilities on finite models, *Symbolic Logic* **41**: 50–58.

Grandison, T. & Sloman, M. (2000). *A Survey of Trust in Internet Applications.*, IEEE Communications Surveys and Tutorials.

Guha, R., Kumar, R., Raghavan, P. & Tomkins, A. (2004). Propagation of trust and distrust, *International Conference on World Wide Web*, pp. 403–412.

Gupta, P. & Kumar, P. (1998). Critical power for asymptotic connectivity, *Conf. on Decision and Control, Tampa*, (USA).

Herken, R. (1995). *The Universal Turing Machine: A Half-Century Survey*, Springer Verlag.

Hubaux, J.-P., Buttyan, L. & Capkun, S. (2001). The quest for security in mobile ad hoc networks, *ACM International Symposium on Mobile ad-hoc networking and computing*, pp. 146–155.

Josang, A. (1996). The right type of trust for distributed systems, *New Security Paradigms Workshop*, pp. 119–131.

Jukna, S. (2001). *Extremal Combinatorics - with Applications in Computer Science*, Springer Verlag.

Kamvar, S. D., Schlosser, M. T. & Garcia-Molina, H. (2003). The eigentrust algorithm for reputation management in p2p networks, *International Conference on World Wide Web*, pp. 640–651.

Kolaitis, P. & Vardi, M. (1987). The decision problem for the probabilities of higher-order properties, *19th ACM Symp. on Theory of Computing*, New York, pp. 425–435.

Kolaitis, P. & Vardi, M. (2000). 0-1 laws for fragments ofexistential second-order logic: A survey, *Mathematical Foundation of Computer Science*, Springer-Verlag, pp. 84–98.

Liagkou, V., Makri, E., Spirakis, P. & Stamatiou, Y. (2006). The threshold behavior of the fixed radius random graph model and applications to the key management problem of sensor networks, *ALGOSENSORS*, pp. 211–223. LNCS.

Liagkou, V., Makri, E., Spirakis, P. & Stamatiou, Y. (2007). Trust in global computing systems as a limit property emerging from short range random interactions, *2nd International Conference on Availability, Reliability and Security (ARES)*, number 0-7695-2775-2, IEEE Computer Society, pp. 741–748.

Liagkou, V., Spirakis, P. & Stamatiou, Y. (2009). Can formalism alone provide an answer to the quest of a viable definition of trust in the www society?, *The 3rd International Conference on E-DEMOCRACY: "Next Generation Society: Technological and Legal Issues"*, Lecture Notes of ICST, Springer.

Nikoletseas, S., Raptopoulos, C. & Spirakis, P. (2004). The existence and efficient construction of large independent sets in general random intersection graphs, *International Colloquium on Automata, Languages and Programming*, pp. 1029–1040.

Richardson, M., Agrawal, R. & Domingos, P. (2003). Trust management for the semantic web, *International Semantic Web Conference*, pp. 351–368.

Spencer, J. (2001). *The strange logic of Random Graphs*, Springer Verlag.

Theodorakopoulos, G. & Baras, J. S. (2004). Trust evaluation in ad-hoc networks, *ACM Workshop on Wireless security*, pp. 1–10.

Trachtenbrot, B. (1950). Impossibility of an algorithm for the decision problem on finite classes, *Doklady Akad. Nauk. S.S.R.* **70**: 569–572.

van Heijenoort, J. (1967). *From Frege to Gödel: A Source Book in Mathematical Logic*, Harvard University Press.

Edited by Felix Jesus Villanueva Molina

It can no longer be ignored that Ambient Intelligence concepts are moving away from research labs demonstrators into our daily lives in a slow but continuous manner. However, we are still far from concluding that our living spaces are intelligent and are enhancing our living style. Ambient Intelligence has attracted much attention from multidisciplinary research areas and there are still open issues in most of them. In this book a selection of unsolved problems which are considered key for ambient intelligence to become a reality, is analyzed and studied in depth. Hopefully this book will provide the reader with a good idea about the current research lines

Photo by MariuszBlach / iStock

IntechOpen

ISBN 978-953-51-5883-7

9 789535 158837