Contributing to energy efficiency through a user-centered smart home

Michele Dominici

ANNÉE 2013

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Européenne de Bretagne

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Informatique

École doctorale Matisse

présentée par

Michele DOMINICI

préparée à l'unité de recherche IRISA – UMR6074
Institut de Recherche en Informatique et Systèmes Aléatoires
ISTIC

Contributing to Energy Efficiency through a User-centered Smart Home

Thèse soutenue à Rennes le 3 juin 2013

devant le jury composé de :

Michel RAYNAL / *Président*
Professeur, Université de Rennes 1
Roberto BALDONI / *Rapporteur*
Professor, Sapienza Università di Roma
Kurt ROTHERMEL / *Rapporteur*
Professor, Universität Stuttgart
Myriam FRÉJUS / *Examinatrice*
Chercheur expert, EDF R&D
Michel BANÂTRE / *Directeur de thèse*
Directeur de Recherche, INRIA Rennes — Bretagne Atlantique
Frédéric WEIS / *Co-directeur de thèse*
Maître de Conférences, HDR, Université de Rennes 1

Acknowledgments

I thank Michel Raynal, Professor at University of Rennes 1, for doing me the honor of chairing my defense jury.

My thanks to Roberto Baldoni, Professor at "Sapienza" University of Rome, and to Kurt Rothermel, Professor at the University of Stuttgart, for accepting to review and judge my dissertation.

I thank Myriam Fréjus, Expert Researcher at Électricité de France (EDF) R&D, for accepting to judge this work.

Special thanks go to my supervisors, Michel Banâtre, Research Director and leader of the ACES team at IRISA / INRIA Rennes — Bretagne Atlantique, and Frédéric Weis, Associate Professor at University of Rennes 1.
Michel, thank you for inviting me to join the ACES team for this research work, for your advice during these years and for your friendliness. Really thanks for believing in me and for offering me the opportunity to exploit the results of my work through a new exciting research prospect.
Frédéric, thank you for your guidance in these years and for your support in the difficult moments — this work would have not been possible without you! Thank you also for encouraging me to improve my dissertation and for guiding me towards a better formulation and presentation.

It was a great opportunity to work at IRISA / INRIA Rennes — Bretagne Atlantique. I thank all the ACES team members and in particular Bastien Pietropaoli, PhD Candidate, and Sylvain Roche, Expert Engineer, for the enriching collaboration and invaluable help during these years. Bastien, you were able to provide elegant solutions for the integration of our two contributions and our discussions about uncertain reasoning were extremely enriching. Sylvain, your technical skills combined with your professional and human qualities made working with you really enriching and pleasant! Special thanks go to Andrey Boytsov, who recently joined the team but participated to the research since almost its beginning, providing help with ECSTRA, his implementation of the Context Spaces Theory. Also thanks to Giulio Zecca, who participated to the first phases of the research.

This work has been supported and guided by an extremely enriching partnership with the personnel and collaborators of EDF R&D. My thanks to Myriam Fréjus (EDF R&D), Julien Guibourdenche (École Centrale de Lyon) and Germain Poizat (Université de Bourgogne / Université de Genève) for providing me with their insights into domestic activity and user-centered design methodology. I really appreciated the efforts you put into explaining me the point of view of cognitive ergonomics on the design of smart homes. In addition, your friendliness made the collaboration really enjoyable. Thanks to David Menga for his extremely valuable technical advice and support throughout the research. Last but not least, thanks to Yves Dherbecourt, Christine Odinot and Pascal Veyret for the very interesting discussions and recommendations.

Finally, thanks to the people that devoted time and efforts to understanding my research problem and provided me with interesting ideas to explore. Between them, I will mention the Triskell team's members (IRISA), Patrick Barlatier and Richard Dapoigny (University of Savoy), as well as John Maraist (Smart Information Flow Technologies).

More personally, I would like to thank all the people that were there for me during these years of work: my family and friends, as well as many of the great people that I met in the research lab and that I now consider as friends.

Table of Contents

Foreword

In 2009, the domestic sector alone represented 30.9% of final energy consumption of electricity in Europe [EEA12b], with a 39.0% increase since 1990 and 6.5% increase since 2005 [EEA12a].

Given the environmental and sustainability issues nowadays, there has been growing interest at the potential for energy saving of households.

Between the stakeholders of such potential energy saving are the electrical utilities. These face growing issues to satisfy a demand alternating between peak and off-peak periods of increasing difference.

EDF, one of the leading electrical utilities in Europe, has led studies towards different perspectives, including human factors underlying consumption. More specifically, the ICAME department of EDF R&D has lead research efforts aiming at identifying "how consumers take consumption decisions and how people could use energy more wisely" [FG12].

On the other hand, the ACES team at INRIA has investigated the use of technology in real-world applications, adopting an approach in which the information and the computations are carried and executed directly by augmented physical entities. This approach allows the integration of computing in the physical world, thus getting closer to people and their environment.

The doctoral research that is presented by this dissertation falls within the framework of a collaborative research project involving the ACES team at INRIA and the ICAME department at EDF R&D. The project aims at combining the insights that emerge from this collaboration in order to enhance the sustainability of household daily life.

M.D.

Introduction

Motivation

In the 1960s, home hobbyists raised the idea of augmenting homes with functionalities offered by Information and Communication Technology (ICT) [Ald03]. Since then, the interest of building, architecture and electronics industries has grown and often led to the inclusion in the design of homes of the enabling technologies, known under the name of *domotics*.

Examples of such technologies are electronically controllable architectural components and devices, like automated shutters, programmable subsystems, like HVAC[1], as well as interconnected devices, like home entertainment systems that play multimedia content stored in other household appliances or on the Internet.

On the research side, numerous projects have explored the potential of technology and sophisticated techniques for the automation of recurring situations and activities of daily living. The result is what is called a *smart home*: "a residence equipped with computing and information technology which anticipates and responds to the needs of the occupants, working to promote their comfort, convenience, security and entertainment through the management of technology within the home and connections to the world beyond" [Ald03].

Despite this growing interest and number of initiatives, the widespread of smart homes seems still far from becoming a reality. It has been argued that the main reason for such a slow uptake is the technology-driven approach that characterizes existing solutions [Ald03]. Such approach has favored the exploration of the technical possibilities over the understanding of people and domestic activity.

A superficial understanding of domestic activity can easily lead to the realization of functionalities that might look sensible but that reveal themselves as unsuitable when applied to real-world households and their daily activity. An example of common assumption about domestic activity is the existence of routines as executions of plans of action. Scientific observation and analysis has refuted such assumption and demonstrated that, although routines exist, they should be seen as recurrent concerns, inherently characterized by variations and irregularities [FG12].

Behind such irregularities, is the fact that the observable behavior of people cannot be separated from their complex "internal state". This is characterized by engagements and concerns, which can be extremely heterogeneous and unrelated to the physical configuration at a given moment.

[1] Heating, Ventilation and Air Conditioning

Unfortunately, a computing system (but also any external observer) cannot capture or recognize such internal state. That is why we talk about a *gap* existing between the sensing capabilities and the reality of human engagements and concerns. To summarize, the point of view of users cannot be captured by a computing system.

Starting from that position, **this doctoral research argues that it is possible to design a smart home that, although not able to capture and understand the user's point of view, can produce functionalities that respect it**. To this end, instead of starting from technological challenges, the design of our smart home has been driven by considerations about real user needs and domestic activity.

Objectives and Challenges of the Thesis

This doctoral research aims at fulfilling three main objectives, which directly follow from the illustrated motivation and approach. We now present such objectives and illustrate the challenges that must be addressed in order to fulfill them.

First Objective

To reach its goal, this doctoral research aimed at **adopting a user-centered interdisciplinary design methodology, combining insights from human factor experts and computer scientists**. We wanted to design future domestic situations as new couplings between home occupants and their environment, enabled by information and communication technology (ICT). This was the first objective of this thesis.

The research was applied to the topical challenge of saving energy at home. For this, the efforts were directed through the joint design and realization of future situations that both promote user comfort and energy saving.

Related Challenges

The design and realization of future situations was done through collaboration between two disciplines as different as computer science and human factor. A challenge connected to this objective has been that of **understanding the point of view of the final users of the system**, as described by human factor experts. This meant understanding what actions of the system might be unsuitable and thus annoying or frustrating for users.

As already stated, what emerged from the interdisciplinary collaboration is that there exists a gap between the capture capabilities of a computing system and the complexity of domestic activity. More specifically, the gap is between the environment as it can be sensed and the environment as it is experienced by people through their activity. Closing this gap asked to **take into account the ambiguity of the observed behavior of people**. Domestic activity is extremely complex and the same actions of people can hide very different purposes and underlying preoccupations, so deciding what actions a technological system might perform is anything but trivial.

Given that comfort and energy saving had to be combined in the design of the future situations, another challenge was that of **intervening into the activity with tar-**

geted actions or proposed actions that would save energy without reducing comfort and possibly even assist the users.

Second Objective

Once the future situations defined, the second objective is to **design and realize a system that provides the functionalities allowing to transform existing situations into sustainable situations**.

Related Challenges

Such objective asks for defining and realizing an effective computing system architecture that can sense and recognize the situations and that can provide the suitable services that transform them. This is enabled by sensing technologies and information-fusion techniques, aiming at capturing and recognizing the *context*, *i.e.*, "any information that can be used to characterize the situation of an entity" [Dey01]. An entity "is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves" [Dey01].

What is challenging in such task, with respect to previous work, is the interdisciplinary process that precedes and follows the specification of computing models. More specifically, the output of the joint design effort is transposed into a technical specification. Starting from that specification, the computing models for recognizing the identified situations are developed. At this point, **the system must offer the tools to (at least, partially) describe such computing models to a non-technical audience, so as to verify that the system is built and behaves as expected**.

For this reason, the modeling tools should allow to represent real-world concepts using intuitive metaphors. Furthermore, the behavior of the system (*i.e.*, the decisions that are made by the system) should be easily understandable.

An additional challenge is generated by the limitations of the sensing technologies and sensor data fusion techniques, which are used to capture information from the environment and to maintain an updated picture of what goes on in the residence. The challenge is that of **dealing with uncertain, imprecise and missing information**.

The degree of imperfection of sensed information must be used to assess the risk of providing unsuitable functionalities, caused by errors in the recognition and decision processes. When the risk is too high, the system should not provide the functionality, so as to avoid bothering users.

Third Objective

Collaborating with psychologists, we identified the constraints to be imposed on the technological choices, being particularly attentive to privacy preservation and respectful of the natural development of domestic life. The aim was to maximize the likelihood that the resulting system will be adopted and accepted in future households.

This resulted in the choice of very simple technologies: we chose to avoid requiring that home occupants carry or wear any instrumentation and we ruled out any privacy-invading technology, such as microphones or video cameras. Thus, **the third objective was that of designing and realizing a smart home that can provide**

adapted functionalities to its occupants, despite relying on a very lightweight instrumentation.

Related Challenges

Being unable to rely on highly sophisticated and ubiquitous sensing technologies has important consequences on the design of future situations and functionalities. The difficulty of recognizing real-world activity becomes even more evident. Additionally, the limitations generated by the adoption of lightweight instrumentation raise the challenge of exploiting to the maximum extent the sensing and actuating capabilities already present in households.

To this end, we chose to bet on the fact that current trends in consumer electronics and household appliances will be confirmed. Indeed, we assist to a process of rapid widespread of highly sophisticated *smart* appliances, which are augmented with sensing and computation capabilities. Examples of such technologies include irons that sense the movement and television sets that can detect the presence of the viewer, so as to turn off automatically when left unattended. This objective raises the challenge of **embedding a context-awareness framework in such appliances and creating a network of appliances that can exchange contextual information, reason about it and exploit their capabilities to provide adapted services**.

Organization of the Dissertation

This dissertation is organized in three parts. In the first one, we argue that the acceptability of many existing smart homes is compromised by the lack of a design process driven by human factor considerations and based on analysis of domestic activity. For this, Chapter 1 presents existing solutions and highlights such drawback. We then present our methodology for the design of smart home functionalities, in Chapter 2, with the main focus being on the interdisciplinary phases of the design.

The second part of the dissertation presents our technical contribution to the research problem. For this, Chapter 3 illustrates existing techniques for modeling and reasoning about context and selects one between them that satisfies the requirements. Then, Chapter 4 defines a system architecture that achieves our goals.

The last part of the dissertation illustrates and evaluates the realization of the smart home. The implementation and deployment choices are presented in Chapter 5. Then, Chapter 6 illustrates the realization of a sustainable situation via an example and evaluates the behavior of the system when facing a realistic scenario. Finally, Chapter 7 concludes the dissertation by illustrating how the realized smart home fulfills the original goal and objectives.

Part I

Present Homes and Future Homes

In this part of the dissertation, we show that the acceptability of many existing smart homes is compromised by the lack of a design process driven by human factor considerations and leveraging a rigorous analysis of domestic activity.

To this end, Chapter 1 presents some existing smart homes and highlights that they are driven by technological considerations, leaving a marginal place to users and the domestic activity.

Then, we illustrate a novel interdisciplinary design methodology addressing the limitations of existing approaches. For this, Chapter 2 extensively describes the different phases that compose such methodology.

Chapter 1

Smart Home: is it only about technology?

The goal of this chapter is showing that most existing Smart Homes have been designed with a technology-driven approach, neglecting human factor considerations. For this, we first introduce some relevant dimensions that can be used to analyze existing Smart Homes. Then, we present few examples of Smart Home projects, following a taxonomy characterized by an increasing level of technical and technological complexity. Finally, we show that increasing complexity does not mean increased usefulness and usability for end users, if the technology is not associated and driven by human factor considerations. We conclude the chapter with the illustration of an alternative design approach and the phases composing it.

Numerous Smart Home solutions have been designed and developed, with two major goals. The first one is optimizing comfort, well-being and quality of life of occupants. The second category of Smart Home systems aims at watching over and assisting elderly and disabled people, in the context of in-home care services.

Given the particular needs of elders and disabled ones, the previous work in the latter category of Smart Homes was often led by considerations about those particular needs. In this sense, we can say that such works took a user-centered (and often interdisciplinary) design approach. However, given the particular target of such solutions, their results are often irrelevant for our research problem. For this reason, we will only present few examples of such category of Smart Homes, just to show their unsuitability for our research.

1.1 Dimensions of the analysis

In this section, we introduce three technical dimensions that can be used to characterize a smart home: *instrumentation*, *context processing* and *exploitation*. The next section will analyze some existing Smart Home solutions with respect to these dimensions.

1.1.1 Instrumentation

The instrumentation is the hardware skeleton of a smart home. It can include sensor and actuator nodes, communication technologies and processing units.

Sensors and actuators Sensors are used to capture information from the environment and actuators are in charge of reflecting on the environment the decisions of the application logic.

Examples of sensor capabilities are user localization and presence detection (using infrared or video cameras, ultrasound or electromagnetic transceivers, *etc.*), temperature, light, humidity and acceleration sensors. Sensors can be embedded in an environment and also be integrated into everyday objects or attached to the human body.

Examples of actuators are lighting and heating devices. *Augmented* (or *smart*) appliances, *i.e.*, household appliances equipped with computational capabilities and communication interfaces, can also be used as actuators.

Communication technologies Communication technologies are used by the pieces of equipment to exchange information and commands. Both wired and wireless links are widespread, exploiting both proprietary and standard protocols.

Processing units Processing units execute the application logic: they periodically or continuously receive and elaborate sensor readings and take the decisions to reflect on the environment through the actuators, either in a centralized or distributed way.

1.1.2 Context Processing

Leveraging the instrumentation, information is sensed from the environment to build a representation of *context*. Using the definition by Anind K. Dey [Dey01], "*context* is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves".

Smart Home systems need to reason about the context information acquired through sensors, for different reasons. Depending on the specific prototypes, reasoning can be used to gain additional information about ongoing situations and human activities, to discover inconsistencies, to learn recurring patterns (used to predict future context evolutions), *etc.*

1.1.3 Exploitation

Results of context reasoning are exploited to provide the functionalities that achieve the user or system goals. For this, the system leverages the processed contextual information to make decisions and exploits the available actuating devices and interaction modalities to realize the functionalities.

In the remaining sections, we present existing Smart Home solutions following a scheme of classification introduced by Christian D. Jensen [Jen], which we extend and modify to take into account recent advances in the field. Jensen divides smart homes in three main categories: *controllable*, *programmable* and *intelligent* houses. The following three sections are devoted to the illustration of each of the categories. In every section, the presentation of existing solutions is organized in accordance with the dimensions that we just presented, *i.e.*, instrumentation, context processing and exploitation.

1.2 Controllable Houses

In a *controllable house*, an occupant can control different devices in more advanced and more efficient ways than it is done in normal houses. Jensen identified three distinct classes of such houses [Jen], illustrated below.

1.2.1 Classes of Controllable Houses

Houses with one integrated remote control In these houses, a centralized control panel or remote control allows to operate several appliances and devices. Wired or wireless infrastructure connects the devices and the control unit. Examples of such controllable houses are commercially available systems for controlling the Heating, Ventilation and Cooling subsystems (HVAC) in a centralized fashion.

Houses controlled by advanced interfaces These houses are able to react to people voice, gesture, *etc.* The goal is to allow a more natural interaction between the house and the occupants, although the underlying functionalities are the same as in the previous category of controllable houses.

Houses with interconnected devices In such houses, several electronic devices (*e.g.*, computers, displays, TV sets, hi-fi systems, additional speakers) are connected with each other. The devices exchange media, resulting in improved entertainment or simplified communication between people.

Figure 1.1 depicts the three classes of Controllable Houses, from left to right. The left-hand side shows a home control panel, able to control lighting, thermostat, security, locks and home entertainment[1]. The central image shows a commercially available product that allows to interact with a gaming console via gesture recognition[2], while the right-hand-side image illustrates devices exchanging media via DLNA standards (Digital Living Network Alliance [DLN]).

Figure 1.1: The three classes of Controllable Houses

[1]Copyright Jan Prucha, 2010
[2]Copiright Microsoft, 2012

1.2.2 Analysis of Controllable Houses

Controllable Houses are mainly the result of efforts of the domotic and home-appliance industries, which commercialize appliances that can be operated through remote controls or control panels. Some research efforts also investigated the potential of using advanced interfaces (*e.g.*, the *Oxygen* project at MIT [Oxy]).

Instrumentation The instrumentation of Controllable Houses is typically made of the same consumer appliances that have to be controlled, with the addition of remote controls or control panels. Communications are realized via wired or wireless connections and often rely on proprietary protocols, even though some standards are available (*e.g.*, KNX[3]). Sensors are typically absent or just provide information about the state of appliances. When advanced interfaces are involved, the sensors may include video cameras or microphones. Actuators are embedded in the appliances themselves and are used to control them. Processing units are absent or limited to what is strictly required to implement the human-computer interfaces. The architecture is often centralized: a central unit processes user requests and pilots the actuators to realize them.

Context Processing and Exploitation No context processing is involved in the operation of Controllable Houses, except when advanced interfaces are involved. In the latter case, video or audio data are interpreted and mapped to commands by dedicated hardware or software subsystems. Since the purpose of the system is simply to allow controlling the home and domotic appliances, the exploitation only consists in executing user commands.

1.2.3 Evaluation of Controllable Houses

Controllable houses improve the way in which different equipment of the house is controlled. However, they are a prominent example of technology-driven smart homes, as the underlying motivation is to bring together the devices present in a house in order to operate them or make them communicate.

For this reason, the design of Controllable Houses often lacks a holistic view based on knowledge and understanding of the domestic needs and activity. Namely, asking for explicit commands and instructions from inhabitants interrupts the course of action characterizing the activity.

To summarize, even though Controllable Houses can provide useful functionalities, these do not seamlessly fit into the domestic activity. This causes user discomfort and is probably one of the causes of the slow uptake of controllable houses. The next category of Smart Homes in the taxonomy is characterized by a different paradigm, where the house can be programmed in addition to being controlled.

1.3 Programmable Houses

Programmable Houses are the second category of the taxonomy [Jen]. Such houses allow programming appliances so that they are switched on, switched off or adjusted

[3] KNX. `http://www.knx.org/`. Accessed on 5 July 2012.

in predefined conditions. Jensen identified two subclasses, illustrated below.

1.3.1 Classes of Programmable Houses

Programmable Houses reacting to time and simple sensor input These houses can be programmed so that some devices are turned on or off at a particular time or depending on the output of simple sensors. For example, lights can be turned on or off at particular times or when a sensor detects movement. In other words, one sensor operates other devices.

Programmable Houses assessing and recognizing situations These houses recognize and react to *situations*, consisting of the combination of predefined conditions, obtained from several sensors. For example, a person is working on its computer and the house turns off the radio to help concentration.

Programmable Houses include mechanisms for processing and combining sensor data, for maintaining and reasoning about context information, for decision making and for automatic orchestration of actuating devices, used to provide functionalities to users.

Several prototypes of Programmable Houses exist. We will provide few examples to illustrate the state of the art in the field. To this end, we analyze such prototypes with respect to the previously identified dimensions.

1.3.2 SM4All project

An example of Programmable House is the outcome of SM4All (Smart Homes 4 All) project, which studied and developed an innovative platform, based on a service-oriented approach and composition techniques, for smart embedded services in immersive environments [AAB+11, SM4]. This has been applied to the scenario of private homes having inhabitants with diverse abilities and needs (*e.g.*, young, elderly or disabled people). The goal of the project was to bring together all devices present in a house and coordinate their activities automatically in order to execute complex tasks that involve many appliances.

Instrumentation In the SM4All demonstration platform, sensors include detectors of motion, smoke and state of appliances. Additional sensors are used to detect brain activity (voluntary electroencephalographic modulations) in the Brain-Computer Interface (BCI) [AAB+11].

Context Processing A rule engine monitors changes in the context and identifies whether certain conditions hold [AAB+11]. When this happens, high level complex goals associated to the identified conditions are triggered. Goals can also be issued by users through advanced interfaces [AAB+11]. These include touch screens and Brain-Computer Interfaces (BCI).

Exploitation High-level complex goals are satisfied by automatically finding the right combination of actuator operations. Examples of goals are preparing a bath, creating a certain mood in a room, following a video, saving energy and closing the house.

Figure 1.2 illustrates an overview of the architecture of the SM4All system.

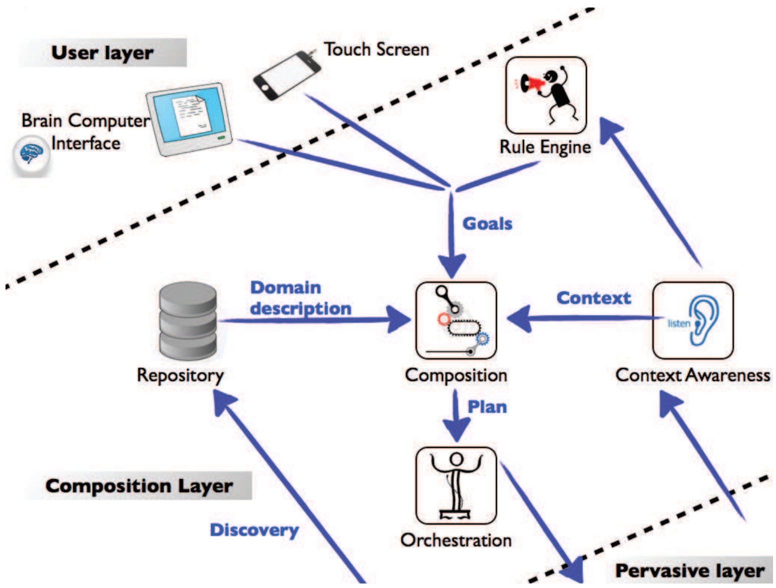


Figure 1.2: SM4All architectural overview (from Aiello *et al.* [AAB+11])

Evaluation of the SM4All programmable house

The SM4All system focusses on the controllability of the house using advanced interfaces and on the coordination of actuating devices. Usability and acceptability of the system have been tested on potential users, with extremely positive results [AAB+11].

However, the main category of users that can benefit from the system are elderly and disabled people, who are disposed to trade off the spontaneity of their behavior for the functionalities that the system can offer. In the reality of the domestic activity of average households, it is often difficult to identify fixed scenarios, corresponding to the illustrated high-level goals and tasks.

For instance, reproducing a predefined mood in a room is a functionality that is rarely suitable for such households. Existing studies show that contextual variability intrinsically characterizes the domestic activity and environment (cf. §2.2). This results in particular needs and requirements that dynamically emerge during the activity and that depend on many factors that vary from case to case. For this reason, predefined configurations of the environment are often unsuitable.

1.3.3 Gator Tech Smart House

The Gator Tech Smart House [HMEZ+05] is a laboratory-house especially designed in order to assist older people in maximizing independence and maintaining a high quality of life. The project's goal is to create assistive environments such as homes that can sense themselves and their residents and enact mappings between the physical world and remote monitoring and intervention services. A supportive and assistive environment for the elderly and the disabled is thus created.

Instrumentation To determine user location, the Gator Tech Smart House has embedded sensors in the floor [AJLS97] and ultrasonic sensors. Each power outlet in the Gator Tech Smart House is equipped with a low-cost RFID reader connected to the main computer. Electrical devices with power cords, such as lamps and clocks, each has an RFID tag attached to the plug's end with information about the device. Smart blinds, mirrors, displays and other devices provide actuating and interaction means.

Context Processing The systems has the ability to abstract state information, interpreting sensory data and identifying high-level states of interest such as "hot" and "sunny" and then carry out actions that correspond to these high level descriptions.

Exploitation Functionalities are selected and triggered based on predefined conditions. Developers visually construct a graph that associates behavior with context; a programmer can also use it to define impermissible contexts and recovery services.

Example of existing applications are: when it is hot, the system turns on the air conditioning; if it is sunny outside and the television is on, the system closes the blinds to reduce glare.

Evaluation of the Gator Tech programmable house

The Gator Tech Smart House focusses on technological and technical issues. The design is not oriented towards implementing functionalities that seamlessly fit in the activity of people. For instance, the system automatically changes the conditions of temperature and illumination of a room on a rule base. There is a great variety and variability of activities that users might perform and numerous other parameters affecting user preferences. For this reason, using a rule-based approach can lead to wrong decisions, which do not take into account all those parameters.

1.3.4 Overall Evaluation of Programmable Houses

Existing Programmable Houses are characterized by rigid characterizations of the conditions that trigger the functionalities. A fixed correspondence between sensor readings and human context is assumed and modeled as rules.

In real-world households, situations are characterized by contextual variability. The same activities can be performed in different ways and the needs of occupants change accordingly. Several other factors can influence user preferences during their course of action and fixed rules often produce unsuitable behaviors of the system. Furthermore, slight changes in the predefined conditions might result in the targeted situations not being recognized.

The next category of smart homes goes one step forward in the adaptation to user needs and preferences, as illustrated in the next section.

1.4 Intelligent Houses

Intelligent houses belong to the last category of the taxonomy [Jen]. Such houses program themselves and their functionalities according to user habits.

The house looks for repeated actions in the daily behavior of occupants. It then programs itself, so as to reproduce user choices (*e.g.*, in terms of switching on or off certain devices) when the identified situations occur.

We will now describe some prototypes of intelligent homes and analyze them with respect to the usual dimensions.

1.4.1 PROSAFE and PROSAFE-extended

The PROSAFE project [CCE03, CC02, CHRC95, SBCC01] proposes a prototype system capable of monitoring elderly or disabled people and of triggering an alarm when a dangerous situation is detected. In a first experimentation [CC02], a prototype system has been deployed in an institution hosting elderly and disabled people. The system is in charge of monitoring patients of the institution during the night, in order to alert the personnel in case of dangerous situations. In a second stage of the project, called PROSAFE-extended [BCE+07, Bon08], another system has been deployed in a residence for elders. In this case, the scenario of application is a whole apartment, where an elderly person lives alone.

Instrumentation PROSAFE-extended exploits infrared wireless movement detectors, a communication box, connected to the telephone and allowing the user and the system to contact a remote assistance center, and a calculation unit, aggregating data and executing the application logic.

Context Processing In PROSAFE, the goal is to identify situations that can be dangerous for the inhabitant. In the first version of PROSAFE, those included falls, runaways and agitation. In extended-PROSAFE, the identification of changes in the subject's physical abilities is also performed, because they could represent a risk. These include reduced speed of motion, reduced covered distance, alteration of the usually followed paths, excessive time spent in the same area of the apartment, and situations in which the inhabitant very frequently gets up, goes to bed or uses the toilet.

Observing the behavior of the users in the previous days, the system uses machine-learning and statistical techniques to calculate a set of thresholds. These include the maximum time that the inhabitant usually spends in a room or in immobility, the minimum and maximum speed of motion or covered distance, the maximum number of times that she visits the toilet, *etc.*

Exploitation The thresholds calculated via machine-learning techniques are used to discern between normal and abnormal behaviors in order to activate an alarm. When an alarm is triggered by the system, the remote assistance center contacts the inhabitant of the apartment to verify the real existence of a danger and then provides feedback to PROSAFE-extended. In this way, thresholds can be adjusted to decrease the triggers of false alarms.

Figure 1.3 illustrates the deployment of the extended PROSAFE system. The picture shows two apartments, on the ground and first floors, equipped with infrared sensors. The calculation units, situated just outside the apartments, collect readings from presence sensors and analyze situations. The communication box in each room is used to alert the remote assistance.

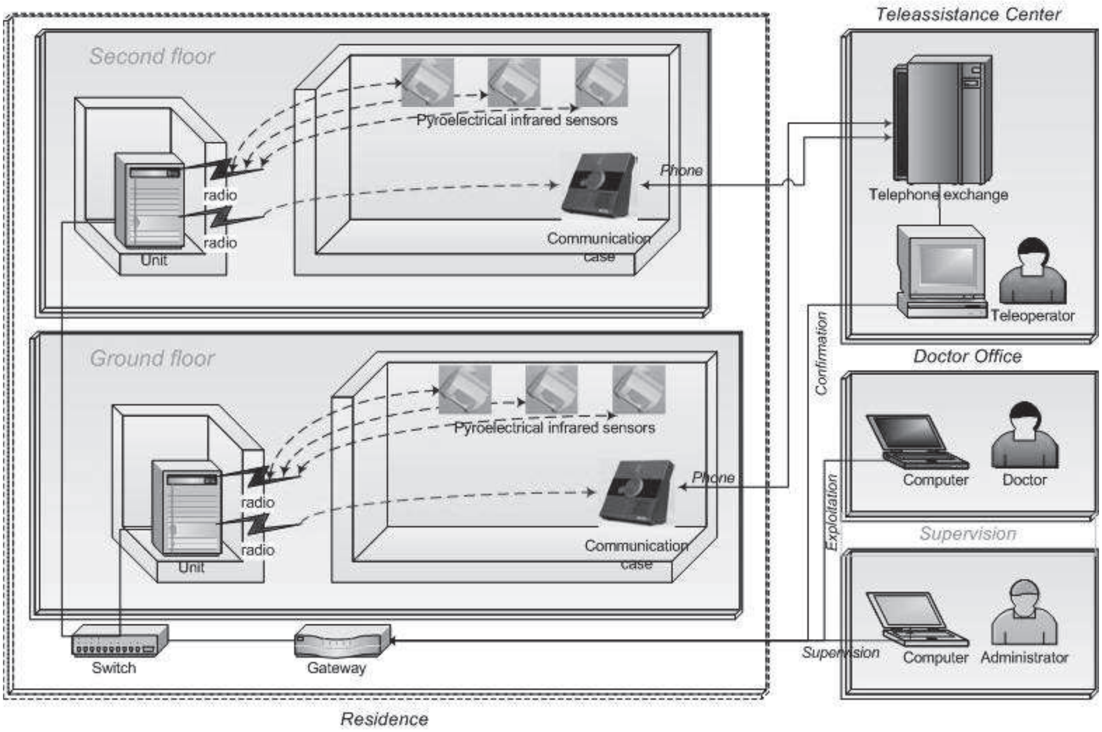


Figure 1.3: The deployment of the extended PROSAFE system (from [BCE+07])

Evaluation of the PROSAFE and PROSAFE-extended intelligent houses

The PROSAFE concept of autonomously monitoring and detecting dangerous situations based on thresholds presents in practice a high frequency of false alarms [BCE+07].

The source of the problem is the impossibility to deal with behaviors that do not follow routines or habits. Raising alarms when the thresholds are exceeded, indeed, assumes that activity strictly follows certain patterns. Such executions of schemes of action do not actually characterize real-world domestic activity, which is instead very variable and basically not predictable with high confidence.

1.4.2 ERGDOM

The ERGDOM project [CDRE+03, Bon08] proposes a solution that manages the house heating system in order to create a comfortable environment for the inhabitants, reduce energy consumption and user intervention.

Instrumentation In ERGDOM, the system exploits two meteorological sensors providing exterior temperature and insolation and a set of sensors placed in each room, including infrared motion detectors and temperature sensors. A central computer executes the algorithms for sensor output acquisition and processing and a simple terminal in each room allows the user to increase or decrease the room's temperature.

Context Processing In ERGDOM, the context is represented by rooms' occupation patterns and comfort temperature. This information is retrieved tracing a history of the motion detections and combining the temperature readings with the choices of the users in terms of increase or decrease in the temperature.

The system learns user habits starting from the history of observations. Those are modeled using the ratio of occupation of each room throughout the day. Based on the probability of occupation of a room at a given moment, the system predicts user behavior, continuously comparing the prediction with the actual situation.

Exploitation An energy efficient control of the heating system is performed considering the prediction of user presence, the comfort temperature, the weather conditions, a thermal model of the house and the energy cost. Figure 1.4 illustrates the functional architecture of the ERGDOM system.

Evaluation of the ERGDOM intelligent house

The ERGDOM system's learning capability aims at avoiding any explicit programming of the heating devices. However, user-satisfaction tests showed that a traditional programmable heating system has comparable operational cost but higher user satisfaction.

The source of the problem can be found in the fact that predictions of room occupation are based on user habits. For this reason, the system is incapable of handling variations in the daily activity timetable.

Real-world domestic activity is highly variable and unpredictable. For this reason, regularities fail in providing a reliable basis on which building the room-occupation prediction mechanisms.

1.4.3 Adaptive House

The Adaptive House [Moz04, Moz98] "programs itself by observing the lifestyle and desires of the inhabitants, and learns to anticipate and accommodate their needs" [Moz98].

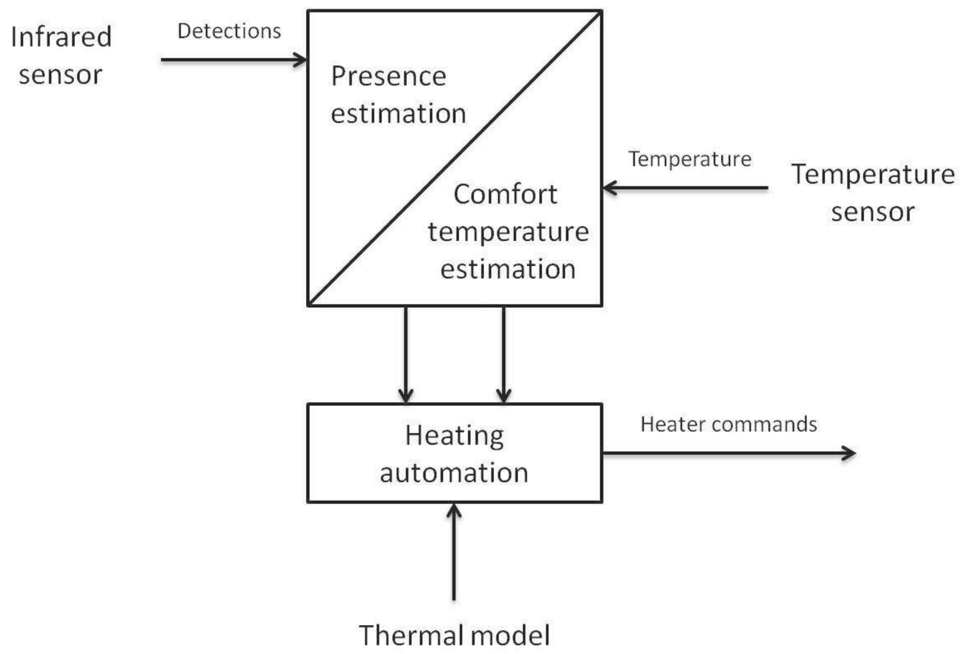


Figure 1.4: The functional architecture of ERGDOM (modified from [CCE03])

Instrumentation In the Adaptive House, seventy-five sensors record various aspects of the environment, including room temperature and light, sound level, door and window openings, motion detections, outside weather and insolation.

The actuators include a whole-house furnace, electric heaters, water heater and lighting and ventilation systems. To regulate lighting, twenty-two independently controlled banks of lights are disseminated in the house, each of which has several intensity settings.

Lights and temperature can be automatically controlled by the system or manually regulated by the user through dimming switches and thermostats. A central processing unit gathers sensor readings coming from the whole house and executes the application logic.

Context Processing The Adaptive House monitors the environment and records the adjustments performed by the inhabitants (*e.g.*, regulating light intensity or the thermostat), observes their occupancy and behavior patterns (exploiting the motion and sound level sensors), and learns to predict future states of the house.

Various predictors attempt to take the current state and forecast future states. Examples of predictions include: expected occupancy patterns in the house over the next few hours, expected hot water usage, likelihood that a zone will be entered in the next few seconds.

Exploitation The Adaptive House has two objectives. One is anticipation of inhabitants' needs, in terms of lighting, air and water temperature, and ventilation. The other objective is energy conservation: lights are set to the minimum intensity required, hot water is maintained at the minimum temperature needed to satisfy the demand; only rooms that are likely to be occupied in the near future should be heated; when several options exist to heat a room, the alternative minimizing expected energy consumption should be selected.

Figure 1.5 illustrates the functional architecture of the Adaptive House. This architecture is replicated for each control domain (lighting, air heating, water heating and ventilation).

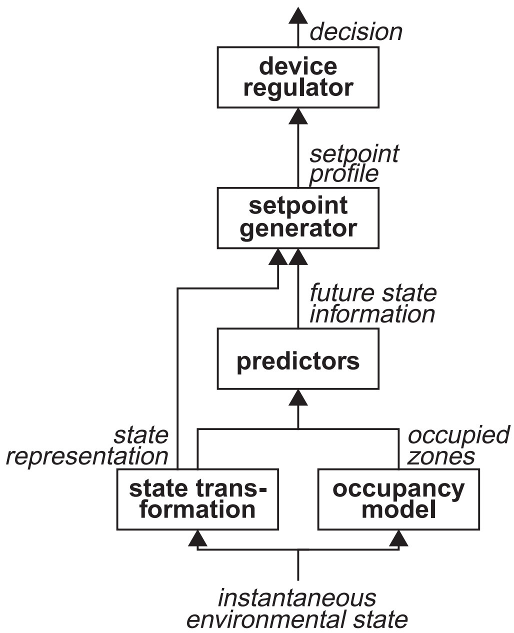


Figure 1.5: The functional architecture of the Adaptive House (from [Moz98])

Evaluation of the Adaptive House intelligent home system

The Adaptive House solution proposes services that help users obtaining the desired level of comfort, reducing their intervention and also addressing the issue of energy preservation. However, tests performed on the prototype showed that prediction errors are frequent and cause high energy inefficiency and user frustration [Moz04].

The source of the problem can be found in the ambitious goal of automatically reproducing user actions. The Adaptive House aims at learning user choices in various situations by observing user behavior. The underlying assumption, thus, is that it is possible to learn and reproduce the decision-making process happening in users' minds based on few environmental cues. The tests confirm that this approach often results in unsuitable functionalities.

1.4.4 MavHome

The goal of *MavHome* (Managing An Intelligent Versatile Home) is to create a home that acts as an intelligent agent [CYH+03, YCH05, CYD06]. The agent's goal is to maximize inhabitant comfort and productivity and minimize operational cost.

The MavHome architecture is a hierarchy of rational agents that cooperate to meet the goals of the overall home. The technologies within each agent are separated into four cooperating layers, as illustrated in Figure 1.6. The *decision* layer selects actions for the agent to execute; the *information* layer gathers, stores, and generates knowledge useful for decision making; the *communication* layer includes software to format and route information; the *physical* layer contains the basic hardware within the house [DCB+02].

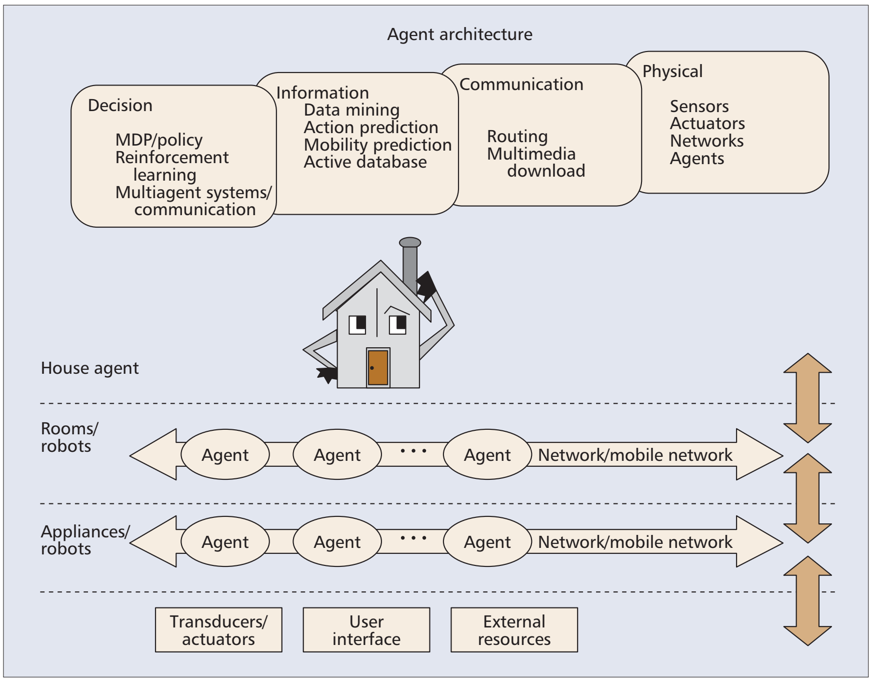


Figure 1.6: The agent architecture of MavHome (from Das *et al.* [DCB+02])

Instrumentation The equipment in MavHome includes power line control interface hardware, touch screens, gesture input devices and cameras [CYD06]. Power line control automates all lights and appliances, as well as HVAC, fans and miniblinds. Perception of light, humidity, temperature, smoke, gas, motion and switch settings is performed through a sensor network. Single-inhabitant localization is performed using passive infrared sensors [YCH05].

Context Processing MavHome harnesses the features of multiple heterogeneous learning algorithms in order to identify repeatable behaviors, predict inhabitant activity and learn a control strategy for a large, complex environment [CYH+03].

The algorithms are first trained by mining sequential patterns from observations of inhabitant activities and interactions with the environment. Next, the inhabitant's upcoming actions are predicted using observed historical data. The interactions of inhabitants with devices during their routine activities are considered as sequences of events with some inherent pattern of periodic recurrence.

Exploitation MavHome learns a decision policy to control an environment in a way that optimizes a variety of possible criteria, including minimizing manual interactions, improving operating efficiency and ensuring inhabitant health and safety. By predicting inhabitant actions, the home can automate or improve upon anticipated events that inhabitants would normally perform in the home.

Minimization of manual lighting interactions was realized in a workplace environment, based on data generated from a virtual inhabitant [CYD06]. The system reduced interactions by identifying patterns consisting in lab entry and exit with light interactions, as well as light interactions at desks.

Evaluation of the MavHome intelligent home system

Tests on inhabitant action prediction showed an accuracy of 86% on synthetic data. However, when applied to real data, the accuracy falls to 30% (only reaching 44% if applied to the recognition of activities contained in previously selected sequential patterns).

Other tests on the percentage of reduced interactions using MavHome showed impressive results (72.2% in the best case). However, these interactions only concern the regular actions of turning on and off lights when approaching and leaving rooms and desks, respectively. This is a very specific and limited aspect of domestic activity, which is much more complex in general. Furthermore, no studies on the acceptability and appropriation of the system have been made.

In real-world situations, fully automatic operation is often unsuitable. For instance, consider the case in which a person enters a room where somebody is sleeping: even though the learned pattern would suggest that the light has to be turned on, what the person will actually do is to leave the light off. In this case, an automatic system turning the light on would be inappropriate and annoying.

1.4.5 Overall Evaluation of Intelligent Houses

Most Intelligent Houses aim at detecting some internal human intent and reacting to it. This is due to the interpretation that most authors do of situations, as mappings between captured context and human context.

Human Context *vs* System Context For instance, if a user is in the study room and accessing the keyboard of the desktop computer, he or she is considered in a 'working' situation [YDM11]. Many applications are based on this interpretation of the human situation: if the sensor data input satisfy the conditions of a situation, the associated application behavior will be executed automatically.

In this doctoral work, we rejected the common assumption that the modeled situation can map between captured context and human context. In our interpretation, situations model correlated context predicates but do not assume any fixed correspondence with human context.

For instance, the situation that sees the presence in the study room and the use of the keyboard would rather be interpreted as 'using computer'. This does not make any assumption about the person's activity or concern. The person might indeed be playing with the computer, or just surfing the Internet, in which case adjusting the sound level of the background music might not be a suitable action.

Full automation A completely automated system that tries to replace users by automatically performing tasks is often not desirable in a domestic setting. Domestic activity is extremely complex and articulated, so automatic systems trying to capture and understand the intents, goals, preoccupations and engagements of inhabitants in order to guess their wishes and execute actions on their behalf are often ineffective. The unavoidable errors of the system are likely to bother inhabitants with the delivery of unsuitable functionalities.

Regularities Many systems currently developed are intended to identify regularities in household activities. However, learning the habits of the inhabitants cannot be simplistically reduced to the identification of extrinsic regularities, ignoring people's concerns. For instance, the ERGDOM system (see §1.4.2) tries to discover regularities in the action of lowering the living room blinds. These patterns are never associated with the actor's intents underlying the action, which may be numerous and different each time. As a consequence, the resulting patterns may not reflect real-world regularities and a system trying to reproduce actions automatically often presents unsuitable behaviors.

1.5 Other works

Existing research in the smart home domain includes numerous other works (*e.g.*, [PBC$^+$03, HMEZ$^+$05, HPB$^+$10]) that we cannot describe here for the sake of brevity. These approaches present similar limitations to those affecting the previously described works.

A notable exception is represented by the *Aware Home Research Initiative* (AHRI) at Georgia Tech. AHRI is "devoted to the multidisciplinary exploration of emerging technologies and services based in the home" [KPJ$^+$08]. It is a research facility that allows simulating and evaluating user experiences with off-the-shelf and state-of-the-art technologies, combining expertise in health, education, entertainment, usable security, computing (HCI, computer vision, activity recognition), electrical and computer engineering, psychology, industrial design and architecture. This is applied to areas like health and wellness at home, sustainability, digital media and entertainment and future tools for the home. In particular, aging in place, busy families and children with special needs are the main themes. The development of the building blocks to create highly distributed sensing and perception technology and to develop awareness of human activity in physical environments is also addressed. The critical resource in this activity is the Georgia Tech Broadband Institute Residential Laboratory (Aware Home), a living laboratory for interdisciplinary design, development and evaluation.

AHRI constitutes an interesting example of multidisciplinary approach to the design and realization of smart home functionalities. The application areas targeted by the research are very diverse. However, pertinent applications for our research are limited to indoor localization exploiting the house power line [PTA06], appliance state detection based on electrical events [PRK$^+$07] and activity characterization using computer vision techniques [KPJ$^+$08].

1.6 The Thesis: A User-centered Smart Home

In this chapter, we highlighted that existing research on smart homes, except for few notable examples, has been primarily focused on the technical possibilities, paying too little attention to understanding the needs of users.

In particular, we showed that many systems aim to model and recognize domestic activity, mapping system context to human context. Given the complex structure of the activity, these systems fail in their goals.

This doctoral research aims at demonstrating that it is possible to design and realize a system that, although not able to capture and understand the user's point of view, can produce functionalities that respect it. To this end, instead of starting from technological challenges, the design of the Smart Home has been driven by considerations about real user needs and domestic activity.

This section provides the background required to understand the approach to the design of a Smart Home that was adopted in this doctoral work. Such an approach has benefited from collaborating with human factor experts. To this end, we first introduce the involved disciplines, followed by a description of the goals and challenges of the design. The approach itself will be extensively described in the next chapter.

1.6.1 An Interdisciplinary Design

The design process is characterized by the cross fertilization between *cognitive ergonomics* and *ubiquitous computing*. We now introduce such disciplines and explain how their combination can allow the design of future situations.

Cognitive Ergonomics

This doctoral research has been characterized by a tight collaboration with experts in cognitive ergonomics. *Ergonomics* is the scientific discipline that deals with "understanding the interactions between humans and other elements of a system" [ERG]. Ergonomists apply "theoretical principles, data, and methods to design solutions that optimize the well-being of people and the performance of the system as a whole". Ergonomists contribute to the "design and evaluation of tasks, products, working conditions and systems to make them fulfill the needs, abilities, opportunities and limitations of human beings" [ERG].

More specifically, *cognitive* ergonomics "is concerned with mental processes, such as perception, memory, reasoning, and motor response, as they affect interactions among humans and other elements of a system" [ERG]. In our domestic scenario, cognitive ergonomics has the role of facilitating and assisting daily life situations [ZF10].

By taking into account the existing constraints of daily life, the ergonomic study of human situated activity allows increased understanding of how behaviors emerge, in connection with the concerns of residents [FG12]. This knowledge can be exploited to take into account people's concerns when designing Smart Home functionalities.

Studies of cognitive ergonomists fed this doctoral research in computer science. The motivation and approach to the research were partially led by ergonomic considerations about domestic activity and acceptability. In particular, the work by Fréjus *et al.* [FG12, GVRG+11, DFG+11, ZF10, PFH09, SDFH09, DSFH05] has laid the foundation, inspired and oriented this doctoral research.

Ubiquitous Computing

When designing the functionalities of a Smart Home, one of the most precious resources to preserve is user attention. During their activities, users should be supported invisibly, reducing interruptions and explicit interactions with the system as much as possible [Wei93]. This observation, raised in 1993 by Mark Weiser, constitutes the foundation of what is known as *Ubiquitous Computing* paradigm.

Following the Ubiquitous Computing (also known as *Pervasive Computing*) paradigm, a number of invisible sensing and computational entities are seamlessly integrated into everyday life, providing adapted functionalities to users, requiring little or no explicit interaction [YDM11].

This doctoral research has followed the Ubiquitous Computing principles and took advantage of existing literature, also leveraging the background in such discipline of the research team in which the research was conducted.

1.6.2 The Goal: Designing *Sustainable* Situations

The combination of cognitive ergonomics and ubiquitous computing creates the opportunity to enhance sustainability in daily life. Considering all the conditions for carrying out the activity (cognitive, social, organizational aspects, *etc.*), we aim to design a new situation that facilitates activity and, at the same time, energy-saving behaviors. We call this a *sustainable situation*.

The objective is a solution that respects the constraints of the household with new adaptive technologies that also allow non-energy oriented functionalities, in order to respond to other criteria such as comfort. This asks for articulating user-centered design criteria (utility, usability, acceptability) with collective criteria (efficiency in household chores, coordination, *etc.*) and with situation-centered criteria (consumption, security) [FG12].

Daily life should be assisted, instead of changed or automated, by designing new situations that help people. From our point of view, the objectives of a Smart Home thus become to: *inform*, *automate* and *ease the takeover*. We now detail such objectives.

Inform Making visible the relevant information to the user according to different contexts (environment and activities), assisting the activity and helping the coordination between inhabitants. Relevant information should be presented to users using nearby media, possibly in the form of non-interrupting notifications (*e.g.*, flashing messages that do not ask for any action of the user).

Automate Performing repetitive and "boring" actions on behalf of the user, like programming a thermostat or turning on and off lights.

Ease the takeover Allowing and facilitating multimodal and ubiquitous takeover. For instance, a smartphone may dynamically change its screenful as the person moves from room to room, so as to allow taking over the nearest appliances. Here emerges the importance of distinguishing situations where the system can execute automated actions from those where it can only assist the user taking over appliances.

To summarize, the functionalities should include providing contextual information to inhabitants, realizing the automatic management of appliances in some specific cases and proposing the remote operation of appliances through advanced interaction modalities.

The provision of contextual information to inhabitants about the state of appliances and other energy-demanding systems facilitates the execution of daily tasks. The automatic management of appliances can reduce energy waste and increase inhabitant comfort by avoiding useless operation and by selecting optimized strategies, especially when considering appliances characterized by inertia (*e.g.*, the heating system). Finally, the advanced interaction modalities allow remote operations on appliances, in the situations in which inhabitants are likely to be helped in their tasks and in effortlessly reducing energy waste.

The goal of designing sustainable situations raises challenges that must be addressed through the cross-fertilization between ubiquitous computing and cognitive ergonomics. The most notable of those challenges is that of filling the *contextual gap*.

1.6.3 The Challenge: Filling the *Contextual Gap*

Providing adapted functionalities necessarily passes by the process of identifying the human context that the functionalities will have to assist. However, the human context itself is not observable, because of its extreme complexity: this is what we call the *contextual gap*. It is the intrinsic discrepancy between the complexity of people's context and the capture and cognition capabilities of a computing system (and of any external observer).

The definition of the situations of interest must thus be transposed to the point of view of the system. The challenge here is how to realize such transposition without losing track of the point of view of the user. Taking into account such point of view, indeed, is strictly required in order to guarantee the suitability and acceptability of the resulting functionalities.

To address such a challenge, we propose to design the Smart Home with an interdisciplinary approach. Such design methodology is illustrated in the next chapter.

Chapter 2

Designing *Sustainable* Situations

As illustrated in the previous chapter, most existing research in the Smart Home domain has mainly focused on technical and technological aspects, neglecting the human factor. This resulted in highly sophisticated systems that fail in providing adapted functionalities to real-world households.

Our research aims at adopting a user-driven design. For this reason, the design and provision of functionalities must rely on studies of domestic activity, so as to understand the way people behave at home. The analysis must also consider people behavior with respect to appliance use, so as to gain knowledge about the sources of energy consumption.

For these reasons, this doctoral research in computer science has been conducted in collaboration with cognitive ergonomists (*cf.* §1.6.1). The adopted technical design and choices respect the constraints, requirements and goals given by ergonomic considerations. To this end, an interdisciplinary design methodology was followed. This chapter illustrates such methodology, describing how we designed the functionalities to be provided by the targeted Smart Home.

In Section 2.1, we provide an overview of the design process. Section 2.2 presents the analysis of domestic activity that was performed by cognitive ergonomists. Section 2.3 illustrates how such analysis can be exploited to design the functionalities of the future Smart Home. Finally, Section 2.4 shows how to realize a paradigm shift from existing to sustainable situations, as well as how to transpose the description of such situations to a system specification. The purely technical phases of the design will be presented in the second part of this dissertation.

2.1 The Three Phases of the Interdisciplinary Design

The interdisciplinary process leading to the design of sustainable situations consists of three phases [HAR98], as depicted in Figure 2.1 and described below.

Phase 1 — Analysis of Domestic Activity and Ergonomic Considerations

Real-world domestic situations and activity are studied by cognitive ergonomists to gain some fundamental knowledge for a deeper empirical understanding of human domestic

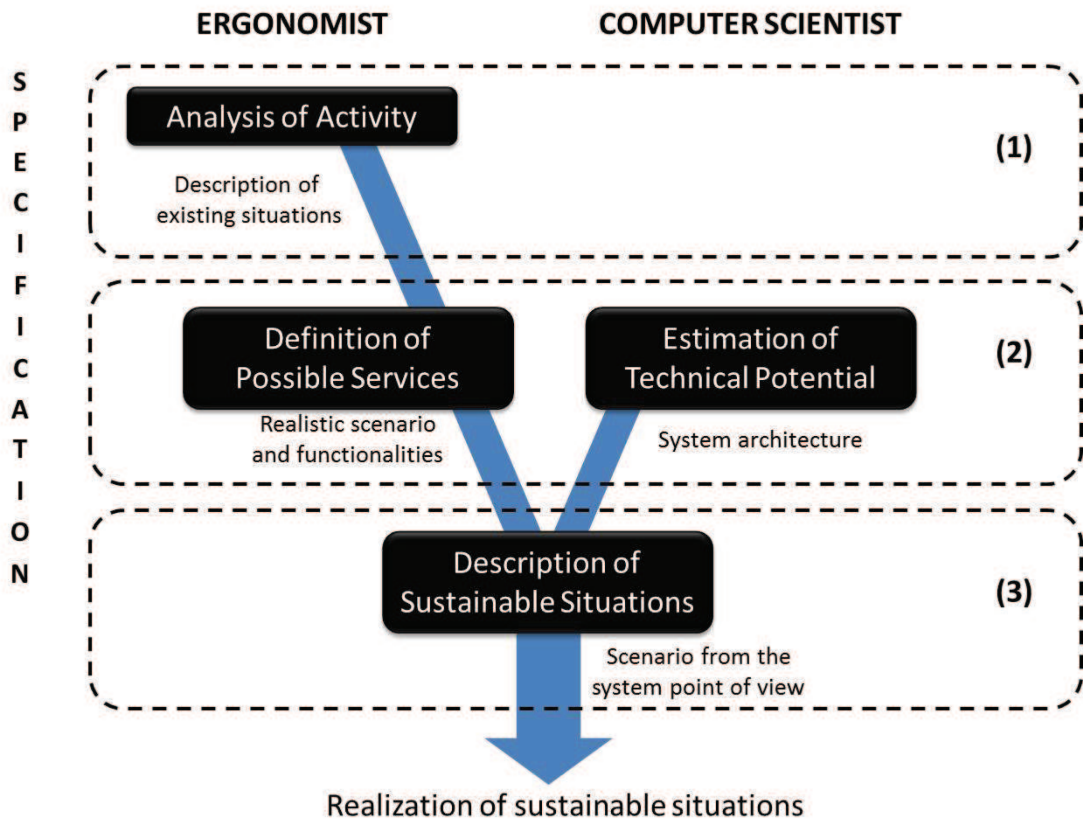


Figure 2.1: The interdisciplinary design approach (modified from [HAR98])

activity. Based on the analysis, heuristic descriptive models of contextual activities are realized. Such process is described in Section 2.2.

Phase 2 — Definition of Possible Services and Technical Potential

Based on the analysis of activity, cognitive ergonomists identify the practices to be preserved and inefficiencies to overcome. Such design phase is described in Section 2.3.

At the same time, the technical requirements emerging from the analysis of domestic activity are studied by computer scientists, with respect to an assessment of state-of-the-art technologies. This results in the definition of a computing system architecture. This process will be illustrated in the next part of the manuscript (*cf.* Chapters 3–4), as it constitutes a purely technical design phase.

Phase 3 — From Existing to Sustainable Situations

In this interdisciplinary design phase, the combination of knowledge of existing practices and targeted services (provided by the ergonomists) with knowledge of technical possibilities (provided by the computer scientists) leads to the description of new, sustainable situations, obtained as improvements of the observed situations. The modalities of human-computer interaction that is needed to fill the gap between the human and the computational context are also specified.

Sustainable situations are described using a realistic scenario of future situations, from the point of view of the system. This design phase is illustrated in Section 2.4.

Realization Phase

Given the outcome of the design process, *i.e.*, a realistic scenario describing future situations from the point of view of the system, computer scientists develop the models and deploy the system that can implement such situations.

The actual implementation of the models and system can help discovering weaknesses in the description of future situations, which can be eliminated through a new design phase, adopting an iterative approach based on feedback. The implementation of the system is described in Chapter 5. The realization of a sustainable situation and the related evaluation, based on the experience of deployment and test, are presented in Chapter 6.

The design approach that we illustrated also enumerates the matters to deal with in the remainder of this chapter. The next section illustrates the approach and outcome of the analysis of domestic activity, which was used as a basis in this doctoral work. Section 2.3 shows how to exploit the characteristics of activity and of household appliances to design possible functionalities. Finally, Section 2.4 illustrates the paradigm shift from existing to sustainable situations.

2.2 Analysis of Domestic Activity

As we previously highlighted, designing the functionalities to be provided by the system should leverage prior analysis of domestic activity. This helps identifying user concerns and needs and centering the design upon a realistic model of individual and collective contexts of activity.

To this end, this doctoral work uses as a basis the analysis of domestic activity that was performed by cognitive ergonomists [FG12, GVRG$^+$11]. We now describe the methodological aspects and the outcome of such analysis.

2.2.1 Methodological Aspects

In the vocabulary of cognitive ergonomists, activity is the set of engagements and preoccupations that characterize a person or a group of people at a given moment. In the case of collective activity, the activity itself can be performed individually in the presence of other people or in a cooperative way.

In order to identify and analyze the engagements and preoccupation of people, a specific methodology must be adopted. In the considered study, the analysis was performed on five dual-earner households with at least one child [FG12]. The data gathered were continuous video recordings of domestic activity and verbalizations during self-confrontation interviews. Each household was recorded six hours a day, during two weekdays and a Saturday. After having agreed, households were recorded in different rooms with synchronic cameras.

To gain a deep understanding of the perspective of the user, the context of the activity was extracted starting from:

- naturally occurring communications and actions;
- interpretations made during interviews by the actors, placed into situation thanks to the video recordings;
- comparing the actions with the environment (state of the artifacts and rooms, co-presence of actors, *etc.*).

First, the data were transcribed with the *multi-score method* [GVRG^{+}11], which allows articulating naturally occurring speech and actions, states of artifacts, physical spaces and interviews' verbalizations. An excerpt of a multi-score transcript framework is presented in Figure 2.2.

In the figure, room scores allow to localize people, to know what they do and what are the different related focus of interaction. Interview's verbalizations can be helpful to detail a particular course of action, while speech and non-verbal actions are detailed next to the actor names. Social interactions are synchronized with each other, across and in the rooms. Simultaneity is on an vertical axis.

The general organization of each individual course of action was reconstructed with the shortened-story methodology [The03], which consists in describing the elements and relations of the situation and activity that are relevant from the user's point of view. Finally, the result was displayed on the multi-score frame again to analyze its distribution at several layers. This methodology was applied to sixty-seven minutes of activity with twelve actors, on four different days in two different families.

2.2.2 Outcome of the analysis

We now present the main characteristics of domestic activity, emerging from the described study and from several other naturalistic studies [BB08, CR04, EG01, GVRG^{+}11, SDFH09, FG12]:

- (Domestic) activity is opportunistic: inhabitants frequently interrupt a particular task for a while in order to accomplish another one.
- Individual activity at home is constituted of multiple lines of different concerns which structure a kind of fuzzy involvement in the activity. For example, a mother can be ironing while following a TV-show and looking after children playing on another floor of the house.
- Inhabitants manage several activities at the same time, with several underlying concerns taking part in their individual context. This is summarized by the expression *cognition and action*: not only concerns, not just physical configurations, but their union. In particular, the same physical configuration may correspond to different situations and a physical configuration does not exclude that other concerns may exist within a person, *e.g.*, I am in front of my computer but I am thinking of putting children to bed.

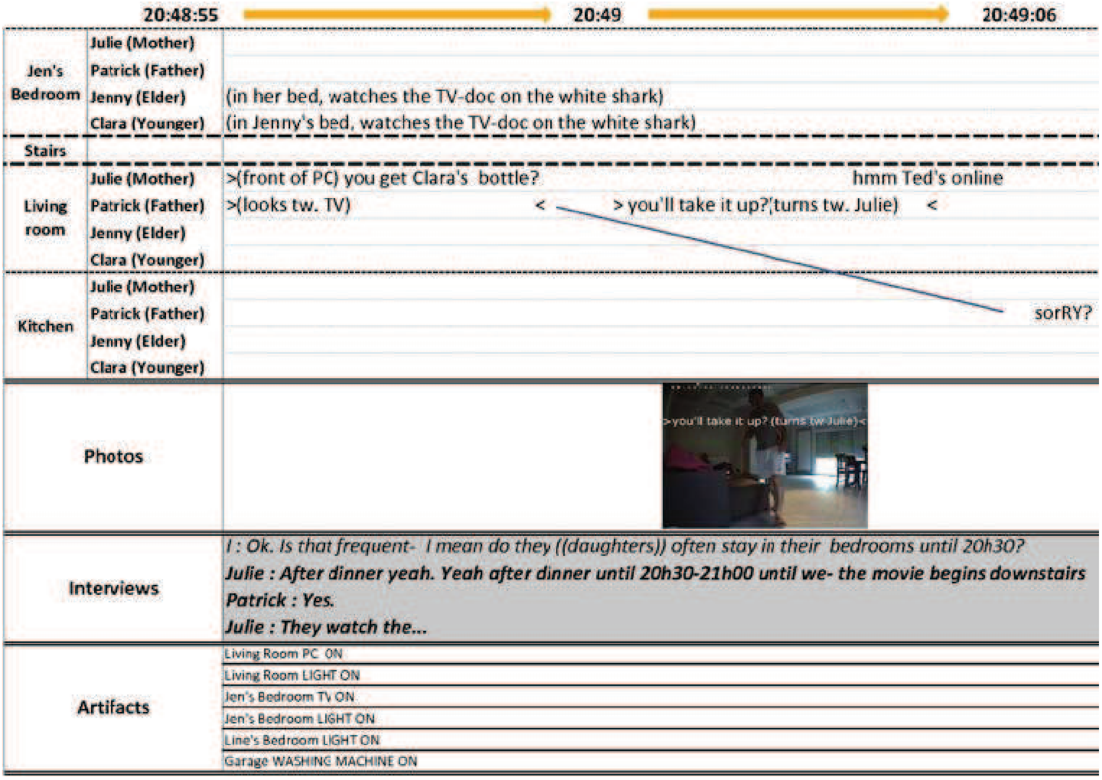


Figure 2.2: Excerpt of a *multi-score* transcript framework (from [GVRG+11])

- Activity is never built according to a pre-established and hierarchical plan but is constantly reoriented according to inter-individual interactions and interactions with the physical environment.

- Routines illustrate the recurrence of concerns (*e.g.*, cooking, taking children to bed, watching night shows on TV), not the execution of plans of action.

- Families are distributed across multiple scales of physical spaces (floors, rooms, systems of tools, voices, noises) and individual and collective scales of activity are intertwined (*e.g.*, house cleaning can be initiated by an individual and finished by another).

- The same behavior, *e.g.*, closing shutters, can have several meanings, *e.g.*, reducing the brightness in a room, ensuring some privacy, increasing the sense of safety, reducing the temperature inside the house.

Such characteristics of domestic activity have important consequences on the design of a Smart Home. The next section shows how to take into account and exploit those characteristics to design suitable functionalities.

2.3 Definition of Possible Services

This section illustrates how the identified characteristics of domestic activity can be exploited to imagine the functionalities of the future Smart Home. The section also shows how to exploit the characteristics of energy-demanding devices that are involved in the activity. Finally, a typical scenario of domestic activity is presented and complemented with the definition of some possible services that limit energy wastage and assist user activity.

2.3.1 Exploiting the Characteristics of Activity

The analysis of activity can be used to identify the practices to be preserved and the inefficiencies to overcome. The following considerations are the result of the work from Fréjus *et al.* [FG12]

Exploiting Recurrent Concerns

As we previously illustrated (*cf.* §1.4.5), regularities in domestic activity exist but should be considered from the point of view of the users, so as to take their concerns into account. The analysis allows to identify in the activity of the inhabitants some significant regularities and regular irregularities.

Significant regularities are recurring concerns, *e.g.*, the household members will eventually put their children to bed. These regularities can be translated into useful *situations* from the point of view of a computing system.

This does not mean that we must learn or detect fine regularities (such as bedtime). It does not either mean that the habits of occupants should be interpreted as executions of schemes of actions. It simply means that we can use the *a priori* knowledge about the recurrence of concerns as a clue that can facilitate the task of recognition of concerns/activities. Based on this recognition, adapted functionalities can be provided. For instance, when observing the recognizable behavior of people going to bed, the Smart Home can switch the heating system to the night program.

Regular irregularities are variations and irregularities inherently characterizing inhabitant routines. Being aware of those irregularities can help developing local interfaces that allow residents to control their environment when the system cannot act automatically.

Understanding the Contextual Variations of the Concerns

Understanding the contextual variations of the concerns allows to identify the contextual state of artifacts or energy waste. For instance, parents of young children often leave lights on in apparently empty spaces such as corridors or rooms, in case children go somewhere dangerous. What might appear as a waste of energy is instead a deliberate choice aiming at improving safety.

On the other hand, parents sometimes leave devices on in empty living rooms or kitchens while caring for children in bedrooms. In this case, differently from the former one, we face energy waste. From the household point of view, however, even this case has a justification, in terms of comfort: parents do not have to switch lights and TV off, so they will not have to switch them on afterward. Comfort here takes the lead.

Transforming the situation here could consist in implementing an automatic system that switches on and off the lights to respond to both the energy-waste issue and the comfort concern. This raises the challenge of correctly recognizing the situation of waste. As we will see (*cf.* §2.4), the last phase of the design has the role of understanding whether the recognition of that situation by a computing system is possible. In case it is not possible, suitable alternatives to automatic actions can be designed, like the mediation of the ambiguous context through interaction (*cf.* §2.4).

Taking Anticipations into Account

There are anticipations that impact the decisions of a smart system designed to reduce consumptions. When parents take children to bed, they are sometimes concerned with what they will do after the children fall asleep: watching TV, reading a book, ironing, resuming diner, preparing for the next day, *etc.* Is it acceptable to switch off a TV downstairs when parents are upstairs in the children's room? They may be waiting for some programs such as weather reports or the beginning of a film.

When designing sustainable situations, taking into account activities that inhabitants might do *later* is important. In the example of the TV illustrated above, a solution might consist in using other media located near the users as a support to allow the parents to follow the beginning of the film until they come back in the living room.

Integrating Local and Global Points of View

The distribution of activity in the house impacts the design of a technical system and the suitability of switching on or off devices. For example, the activity of caring for children requires an important engagement and interaction between parents and children. Hence, parents are barely able to actually follow what is going on in other rooms. In this context, switching off the other rooms' lights and TV would surely be the relevant output of the system. The context is provided by multiple activity and environmental cues that seem to argue for switching off.

However, there are cases where only one parent goes upstairs for taking children to bed. The other parent stays in the living room, following the TV show while quickly moving in and out of the kitchen. This shows that taking children to bed do not emerge the same way at the scale of collective activity and at the scale of the house. In this case, the context provided by the multiple cues seems to argue for not switching off.

Considering the Spatial Distribution of People

When designing Smart Home functionalities, there are also ambiguous issues resulting from the contextual meaning of concerns and the spatial distribution of people in the home. 'Taking children to bed' does not begin by coming in the bedroom. Many times it begins outside the bedrooms, for example when parents and children 'negotiate' the organization of the evening. Preparing a bottle for children mobilizes kitchens for example, not only bedrooms. A clear and fixed association between concerns and rooms seems irrelevant for the design. In this case, environmental cues seem to play only as negative definitions in that they only allow determining what is *not* going on for people.

Ergonomists observed that a TV may display programs without being watched by people in the same room, implying that one might switch it off. On the contrary, the TV sound often provides accounts in the rooms nearby that allow people following programs at a relative physical distance from the living room. This ambiguity leads to envisage the potential design of explicit interactions between household and the system, in order to decide whether they want to switch off or not.

This section presented how to exploit the characteristics of domestic activity to imagine adapted functionalities that take into account the concerns of people. We now continue by analyzing the characteristics of household appliances that can help designing Smart Home functionalities.

2.3.2 Exploiting the Characteristics of Appliances

We present here some dimensions that characterize household appliances. Based on the identified characteristics of appliances, designers can conceive adapted functionalities. These can consist of advanced management strategies for the appliances or of contextual services provided to users by the augmented appliances themselves.

The criteria are meant to inspire the design and not to constitute formal specifications. Additional ergonomic considerations have to be formulated so as to guarantee that the resulting functionalities are adapted to the situations from the point of view of users.

We first provide a description of the dimensions of the analysis, followed by a set of examples describing common household appliances. The following considerations constitute an original contribution of this doctoral work.

Need for Manual Intervention The *need for manual intervention* criterion specifies the needs of the appliance in terms of manual operation by people. The appliance can be autonomous, ask for procedural (*e.g.*, on/off commands), periodical or continuous interventions by a human. The knowledge of this characteristic of the appliance can be used to make actions influencing its operation. For example, an appliance asking for frequent interventions will help predicting the presence of somebody. The need for manual intervention of the appliance can also depend on context.

Need for sensory presence The *need for sensory presence* criterion specifies if someone must be able to hear, see, touch or feel the effects of the appliance for it to operate purposefully. For instance, a television is useless if nobody is able to view or hear it, so it can be turned off in that case.

Impact on domestic environment The *impact on domestic environment* criterion specifies the effects of the operation of the appliance on the domestic environment. The effects may be the produced noise, in the case of a washing machine, or sound, in the case of a television. Both noises and sounds are important with respect to the domestic activity, *e.g.*, the noise of the machine can be used by people to know when the washing has finished.

Functionalities might be conceived based on the information about the impact of an appliance. For instance, a notification can be delivered to users to inform them about the end of the washing and an audio channel can be established between two rooms when the noise covers the conversation.

Augmented functionalities The *augmented functionalities* criterion indicates the functionalities that the appliance might offer when augmented with communication and computation capabilities. Computation capabilities allow the appliance to receive contextual information from other appliances and/or processing units. Computation capabilities enable the appliance to reason about context and situations to adapt its behavior accordingly.

Provided evidence The dimension of *provided evidence* specifies what kind of clues about user activity or situation can be inferred from the use of the appliance. For instance, if the iron is being moved in a room, somebody is present in the room and is ironing.

Linked appliances *Linked appliances* are other devices that, if connected together, might improve the functionality of the augmented appliance. For instance, if a smartphone is connected to a hi-fi system, the devices can cooperate so that the hi-fi does not play loud music during a phone call.

Contextual information that may improve operation The *contextual information that may improve operation* indicates what contextual information might help the appliance making better decisions. These concern its own functioning strategies and the interaction with users or other devices. For example, a smartphone knowing its location and surrounding devices can be used as a contextual remote-control that allows a person to switch on the light when entering a room.

Figure 2.3 shows few examples of analysis of common appliances based on the previously illustrated criteria. Each row of the table analyzes a common household appliance with respect to those criteria. For instance, a Hi-Fi system is characterized by a procedural interaction, *i.e.*, it has to be turned on and then off. Then, it can function autonomously, provided that someone is able to hear the produced sound. An iron, instead, must be handled continuously, in order to justify its operation; otherwise, it can be turned off. Functionalities can be straightforwardly designed based on such analysis.

2.3.3 Outcome of the Design Phase

The presented characteristics of activity and appliances are used by cognitive ergonomists to develop a typical scenario of domestic activity. An example of such scenario is presented in Figure 2.4.

Appliance	Basic functionalities	Need for manual intervention	Need for sensorial presence	Impact on domestic environment	Provided evidence	Linked appliances	Augmented functionalities	Contextual information that may improve operation
Tablet	Information & multi-media content provision, communication, text-editing, *etc.*	Depends on application (*e.g.*, *continuous* if pdf reader application runs in foreground, *autonomous* in case of music player running in background)	Context-dependent (*hearing* if the running application is music, *hearing* and *view* if playing a video, *etc.*)	Can produce sound, issue sound notifications	Presence (when operated)	Any	Provide information, act as input device	Presence, localization of the appliance, *etc.* (depending on application)
HVAC	Heating, ventilation, air conditioning	*Procedural* (on/off command)	*Feel* the temperature	Big amount of produced / absorbed heat	Presence and feedback through direct operation	Any appliance producing or absorbing heat	Finely tune and anticipate operation depending on thermal model and other running appliances	Forecast about energy use, price and about occupancy and activity. Thermal model of house and rooms.
Smartphone	Communication, entertainment, *etc.*	Depends on application (*e.g.*, for phone call: before and after the call)	Depends on application (*e.g.*, *hearing* when making phone call)	Can produce sound, issue notification	Presence and activity (using embedded sensors)	Any	Provide information, act as input device, act as remote control	Localization and state of other appliances, so as to contextually change the interface to control the surrounding environment
Hi-Fi system	Audio diffusion	*Procedural* (on/off command)	*Hear*	Produces sound	Presence (when operated)	Noisy appliances: to propose to turn something OFF	Turn off when left unattended (detected combining presence + presence in adjacent rooms)	State of appliance producing loud sound/noise, level of ambient noise, localization
TV	Multi-media content	*Procedural* (on/off command)	*View* or *Hear*	Produces sound	Presence (when operated)	Lights, noisy appliances	Turn *screen* off when nobody can see, propose to create ideal lighting conditions, propose to remotely control noisy appliances	State of appliances producing loud sound/noise, level of ambient noise, localization
Washing machine	Washing clothes	*Procedural* (load + turn on + unload)	None	Produces noise during particular phases	Presence (when (un)loading), future presence (washing finished)	Lights, devices needing *hearing*	Notify when finished, propose to postpone noisy cycles when appliances needing *hearing* are on	State of appliances needing *hearing* sensorial presence, localization
Iron	Ironing clothes	*Continuous* (except while waiting to reach temperature)	None	Small amount of produced heat	Presence (when being moved)	Lights	Turn off when left unattended, propose to adapt lighting	Localization: to control the suitable lights

Figure 2.3: Analysis of common appliances

Time	Actions	Concerns /	Characteristics of the activity	Automatic energy management
4pm	The parents take advantage of their time off on Wednesdays to do housework, as usual. The older child is at day camp (no school on Wednesdays). The younger child (a toddler) is watching TV in her room.	Catching up on housework. Watching over the younger child.	Recurrent activities, each with a specific duration (housework, home repairs, washing dishes, *etc*.). Routine activity but carried out a bit differently each time.	The younger child's room is heated to the comfort level. The older child's room has a lower temperature.
	The father collects laundry to be ironed or put away in the bedroom. The mother makes tea in the kitchen.	Father: taking care of laundry. Mother: taking a break.	Individual activity is part of the collective activity. Local activities are part of a more global involvement.	Lights on in parents' bedroom (if needed, depending on natural light), the rest heated to mid-level temperature (except the older child's room). Bedroom and kitchen: heating (or air conditioning) set to comfort.
4:11	The father is going to tidy up in the living room. The mother uses the vacuum in the entryway.	Father: picking up in the living room. Mother: vacuuming.	No strict association: one activity/one location.	Lights on in the living room and entryway/ heating to comfort. Vacuum: all lights on. Parents' bedroom and kitchen change to mid-level temperature heating after 15 minutes.
4:20	The father takes advantage of being near the office to check his e-mails. The mother vacuums in the living room.		Interruption/ opportunistic activity. Collective activity can be "crossed" by individual activities.	The office heating changes to a comfort setting, the living room as well; the kitchen changes to mid-level temperature.

Figure 2.4: Typical domestic scenario and possible functionalities

The scenario also includes hints for the design of adapted energy-saving services. However, such indications do not provide technical specifications concerning their implementation. To reach such specifications, an additional effort must be undertaken, characterized by a tight collaboration between computer scientists and cognitive ergonomists. Such phase is illustrated in the next section.

2.4 From Existing to Sustainable Situations

Going from the description of the activity to the specification of the future application asks for a paradigm shift [HAR98]. From the process of describing an existing situation, one has to switch to the process of designing a situation that does not yet exist.

Ergonomists or computer scientists alone cannot predict what the future situation will be. The combination of the knowledge of existing practices (provided by the ergonomist) with knowledge of technical possibilities (provided by the computer scientist) will help guiding the specification process towards adapted functionalities.

We already presented the issues raised by the gap between the complex human context and the context of the system based upon sensing (*cf.* §1.6.3). A computing system can only detect and recognize the raw physical phenomena characterizing the environment. The challenge is thus to take into account this gap in the design process and in the interaction definition.

In this section, we first illustrate how to design the interaction required to fill this gap (*cf.* §2.4.2). Then, we present the outcome of the design phase, a realistic scenario of future sustainable situations, from the system point of view (*cf.* §2.4.3).

2.4.1 Filling the Contextual Gap

In the previous chapter, we criticized the functionality consisting in automatically adjusting the level of the background music when a person is using a computer (*cf.* §1.4.5). In the illustrated example, we said that a system should *not* act automatically in that situation.

The reason is that, acting automatically, the system does not take into account the contextual gap. Indeed, sensing that a computer is being used cannot be taken as a complete determination of user context. Decreasing the background music is a suitable action only in some particular cases. And only knowing that the computer is being used is not enough to guess if the current situation of the user requires such action.

The reader might ask what should *instead* be done to assist the activity. In that specific example, the answer is in the interaction between the system and the inhabitants. An automatic action of the system is not suitable, because it would require knowing the real intention, engagement and concern of the person. What we consider to be suitable, instead, is proposing the person a remote operation on the Hi-Fi system. So, instead of automatically adjusting the sound level, the system may use the desktop computer that the person is using as an interface to propose the functionality. A notification icon might unobtrusively be displayed on the screen, allowing to adjust the sound level with a single-click interaction.

To summarize, interaction with users is sometimes used to fill the gap between real-world situations and recognition capabilities of the system. Thus, adapted functionalities can be proposed via specifically designed interaction modalities, so that inhabitants are provided with targeted propositions that minimally interrupt their activity. Such process is illustrated in Figure 2.5.

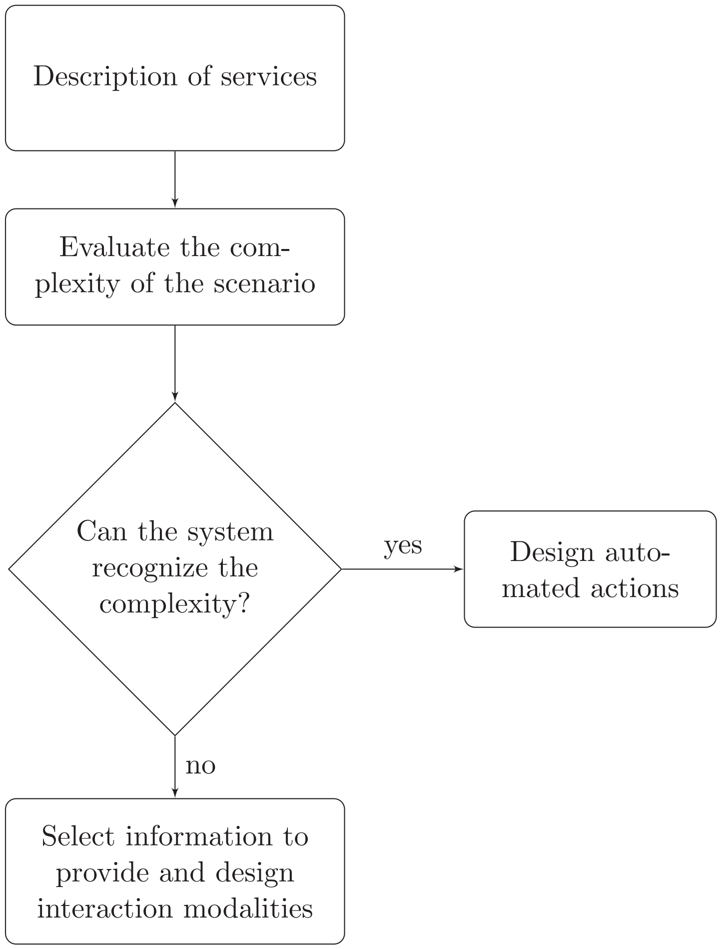


Figure 2.5: Filling the contextual gap

2.4.2 Designing the Interaction

We highlighted that the contextual gap can be filled through adapted interaction strategies. We now provide few principles that can guide the design of such interaction modalities.

Validating Validation can be performed using notification mechanisms that allow the inhabitant to regain control of the system before an unwanted action is performed. For instance, before turning off the television after reaching the conclusion that nobody is watching it, the television could display a warning message informing of the pending action. If a person was there without being detected, he could stop the imminent action. Obviously, this process of interaction should not occur too often to avoid bothering the person.

Delegating The alternative consists in allowing users to deliberately put a zone of the house in automatic mode. When a room is put in automatic mode, all the appliances located in the room are automatically managed by the system, in spite of uncertainty. Adapted modalities allowing the user to select between automatic and manual mode should be conceived.

Detecting oversights The interaction with the user may be triggered whenever an oversight is suspected. For example, a timer could watch over the duration of the absence from the kitchen while a hotplate is running, so that the oversight can be inferred after some minutes. The recognition of oversights could be helped by information about ongoing activities of the actors. For instance, knowing that the person is busy watching the television may suggest that the hotplate has been forgotten. In that case, the television can be used to warn the user about the oversight. It can also be exploited as a medium to remotely operate the unattended appliance.

2.4.3 From the User Point of View to the System Point of View

We provided few hints about how to design the interaction strategies, which are required to fill the contextual gap. The challenge that emerges now is harmonizing what is produced by ergonomists and computer scientists during the specification phase. The goal is not to impose a common language for both disciplines, but to agree on the fundamental principles that will guide the realization of the Smart Home functionalities.

A solution to such a problem is obtained by building an intermediate scenario of activities, a collective construction that is understandable by both cognitive ergonomists and computer scientists. It is a scenario describing the future sustainable situations, resulting from the seamless interaction of the Smart Home and the occupants.

The scenario describes a sequence of actions that the system performs in response to the observed behavior of users. Although the scenario describes the situations from the point of view of the system, the joint work with ergonomists guarantees that the key characteristics of activity and human cognitive functioning are taken into account and respected.

An example scenario that results from this design phase is illustrated below. This scenario is designed for a three-room residence (one bedroom upstairs, kitchen and living room on the ground floor), occupied by two people and equipped with an electric heater in each room.

1. The scenario begins in the morning: two of the rooms are unoccupied (the bedroom is occupied) and the two heaters are set at a minimal level (*comfort-2*). Heating starts in advance of waking and rises to an average level (*comfort-1*) on the basis of the wake-up time set on the augmented alarm clock.
2. The first person enters the kitchen and his presence is detected. If the presence is maintained and activity is still detected by the system, the room heating rises to the comfort level.
3. The second person enters the living room. Similarly to what happens in the kitchen, the heating rises to *comfort-1* and then to comfort level in case of prolonged activity. This person then goes into the kitchen. If the system calculates the lack of activity in the living room to be beyond a timeout value, it decreases the heating level by one level of comfort.
4. The stovetop heating elements in the kitchen are activated by a person. Nothing happens as long as activity remains in the kitchen. In contrast, if an extended absence is detected in this room (in our scenario, the two people are now in the living room), the option of turning off the stovetop is offered to the user through a tablet in the living room.
5. The radio is on in the kitchen and playing music. The phone rings. At this point, two cases can be highlighted. If no activity is detected in the kitchen, the sound coming from the radio is not considered as a nuisance to the inhabitants and the system does not intervene. If the kitchen is occupied, however, the phone offers an adapted interface for muting the radio.

The presented scenario illustrates the behavior of the future system in response to observed situations. It can be used as a basis to design the computational models of such system. These will allow the recognition of the targeted context and situations, as well as the provision of the functionalities. The design of the computational models required to implement this scenario will be presented in Chapter 6.

The remainder of this chapter evaluates the result of the design process and provides fundamental considerations for the design of the technical aspects of the Smart Home.

2.5 Evaluation of the Resulting Situations

The interdisciplinary design process that we described culminates in the description of future sustainable situations, from the system point of view. This has been presented in the previous section through a scenario.

Such scenario contains measures to take into account the impossibility to observe the user's context. This section highlights these measures and their effectiveness.

2.5.1 Taking Engagements into Account

In the illustrated scenario, the strategies for managing the heaters show that the system takes into account the possible engagements of people. In the scenario, people move between the kitchen and the living room (*cf.* paragraphs 2 and 3 of the scenario). Each time a person enters a room, several engagements are possible: "I am coming in" *vs.* "I am coming in and then leaving" *vs.* "I am going to leave but somebody will be coming in". Obviously, the system is unable to identify the ongoing one. Thus, it is not possible to select the most suitable action to perform on the heaters.

To fill the gap between the observable conditions and the real user context, we leverage time intervals. The time intervals between observations (*i.e.*, a person entered a room) and automatic actions (*i.e.*, changing heater state) enable to decide which is the most probable engagement. In other words, the time intervals allow to accumulate evidence to identify the unobservable user engagement.

To summarize, future situations involve automatic actions of the system, like managing the heaters, but also strategies to take into account the possible engagements of people, so as to fill the contextual gap. To this end, the temporal dimension is a key decision-making criterion, allowing to maximize the likelihood of providing a suitable functionality.

2.5.2 Reacting to Undecidability

The scenario includes a strategy to leave the choice to users. More specifically, the action of muting the radio when the phone rings is not performed automatically (*cf.* paragraph 5 of the scenario).

Instead, an adapted interface to mute the radio when answering the phone is provided. This has the main advantage of allowing the user to decide if the phone call and the radio are incompatible in that particular situation. An automatic decision is impossible in that case, because the answer may depend on the occupation and number of people in the room and may also require a negotiation phase between occupants to make a decision.

So, delegating the choice to users has the advantages of remedying to sensing limitations (*e.g.*, the inability to determine the number of people in a room) and of allowing people to express their preference.

Furthermore, the interaction modalities can be designed so as to minimally interrupt the user's course of action. For instance, the choice of the preferred action can be made when answering the phone, exploiting a contextual incoming-call screen. This screen allows switching off the radio while answering the call, as showed in Figure 2.6. By performing two actions while pressing a button, the application requires a single interaction with the resident. Furthermore, that interaction would anyway have been necessary (to answer the phone call), so the application does not interrupt or disrupt the normal course of action. The resulting functionality assists the activity, increases comfort and saves energy[1].

[1]The described functionality was actually realized in the context of this doctoral work and runs on common Android smartphones.

Figure 2.6: Contextual incoming-call screen

2.5.3 Anticipating

In the scenario, an augmented alarm clock is exploited to infer the expected wake-up time (*cf.* paragraph 1 of the scenario). This is used to automatically manage the heating system, saving users the trouble to manually program it.

This example demonstrates the strategy that consists in exploiting available clues to infer future situations. As we said, routines are just the recurrence of concerns and we cannot expect the wake-up time to be the same every day. Exceptions and irregularities inherently characterize the activity (*cf.* §2.3.1).

Thus, we consider the alarm time to constitute evidence of the future wake-up time. In case the house occupants set a morning alarm, this information will be provided by the augmented alarm clock and used to manage the heaters accordingly.

In this way, we do not rely on regularities, which may lead to wrong conclusions, but we accumulate evidence originating from spontaneous actions of users.

2.6 Conclusion

In this chapter, we described our interdisciplinary approach to the design of sustainable domestic situations. These are new couplings between the house and its occupants, where the activity is assisted and transformed through seamless interaction between them, with the goals of improving comfort and saving energy.

We illustrated that realizing a Smart Home undergoes a design process rooted in the analysis of domestic activity and guided by ergonomic and technological considerations. As explained in the previous chapter, existing work either empowered residents to control their house or developed systems that replace user actions after having recognized recurring patterns.

Our approach, instead, showed how existing situations can be modified through technology, in order to optimize some targeted criteria, like comfort and energy consumption. To this end, we described the way computer scientists can collaborate with cognitive ergonomists in a joint design effort, where different points of view and expertise converge towards the realization of sustainable situations.

More specifically, cognitive ergonomists can help computer scientists gaining a deeper understanding of human domestic activity. On the other hand, computer scientists can contribute to the design of future sustainable situations with their technological expertise, by providing an assessment of the technical potential. This achieves the cross-fertilization between disciplines.

An intermediate collective construction is used to enable an agreement on the working principles of the future system. It is a scenario that describes the future situations from the system point of view and that is understandable by both cognitive ergonomists and computer scientists.

The next steps in the process of realizing our Smart Home are the design of the technological system and of the computational models that enable the realization of the conceived situations.

The next part of this dissertation will illustrate the contribution of this doctoral research to such problems. Such contributions include the design of the system architecture and the realization of a prototype system that enables the sustainability of domestic activity.

More specifically, Part II will identify the requirements and define a context-aware system, while Part III will present the implementation, deployment and evaluation of a prototype.

Part II

Building a Smart Home

The previous part of the dissertation described the interdisciplinary methodology that we used to design future domestic situations, resulting from a new coupling between the house and its occupants. We now illustrate and solve the technical challenges involved in the design of the Smart Home that enables such sustainable situations. This constitutes the technical contribution of this doctoral research.

For this, we first describe a fundamental capability on which rely the provision of Smart Home functionalities: the *context awareness*. To this end, Chapter 3 presents existing techniques for modeling contextual information and reasoning about it, in order to infer the occurrence of interesting situations. Such techniques are evaluated with respect to their suitability to the goals of this doctoral research. The chapter also identifies a suitable technique that will provide the basis of our contribution.

Based on these foundations, Chapter 4 describes the architecture of the context-aware system that we defined and developed. Such system augments the house with the 'smartness' that is required to enable the future sustainable situations.

Chapter 3

Being Aware of Context

The previous part of this dissertation illustrated how we conducted the design of sustainable situations, demonstrating that the functionalities provided by a Smart Home can take into account the point of view of the user. We now analyze the technical characteristics of a system that provides such functionalities, in order to realize the targeted sustainable situations.

For this, we need to introduce a fundamental capability that allows the provision of Smart Home functionalities: the *context awareness*. A system is *context aware* if it "uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task" [Dey01].

When applied to the Smart Home domain, context awareness allows the provision of adapted functionalities to occupants. For example, under some conditions, knowing that an occupant has spent long time in a room can suggest that the system should turn on the room's heating and turn off the other rooms' lights.

In order to design a context-aware Smart Home, we need to review the literature of context-awareness techniques. This allows us to leverage existing work and know-how, so as to bring our contribution to the knowledge of the domain. However, context-awareness has been a hot research topic since numerous years and an extensive review of existing approaches is out of the scope of this dissertation.

This chapter has three less ambitious goals. The first one is to highlight the lessons that can be learned from existing literature. This is the role of Section 3.1.

The second goal of this chapter is to identify the requirements of the context-awareness techniques to be used in our Smart Home, as well as to present and evaluate some major existing techniques. This is done in Sections 3.2 to 3.5.

The third goal of the chapter is to extensively present and evaluate the *Context Spaces Theory*, a context-awareness framework that fulfills many of our requirements. We do this in Sections 3.6 and 3.7. Such theory will provide the context-aware capabilities of our Smart Home, as we describe in Chapter 4.

3.1 Overview of Existing Context-Awareness Techniques

In this section, we first present how context-awareness techniques have evolved from data-centered to situation-centered. Then, we illustrate the main characteristics of existing situation-awareness techniques.

3.1.1 Evolution of Context Awareness: from Raw Data to Situations

Since its early age, research on context-aware systems has evolved from a data-centered approach towards a situation-centered approach. We now illustrate this evolution, describing the reasons behind it. As we will show, situations are convenient and intuitive modeling concepts that simplify the design and use of context-aware systems.

From Data-Centered to Model-Centered Approaches

Context-awareness relies on the capture of contextual information from the environment, mainly achieved using values provided by sensors. Early research on context-aware systems used those values to directly build applications.

Unfortunately, the interpretation of sensor data may be complex, because of the volume of data, their heterogeneity and inter-dependence [YDM11]. Furthermore, sensors produce noisy data, affected by uncertainty and imprecision. This is due to technical limitations and to communication or sensor failure. In addition, directly using sensor data in applications complicate their implementation and reuse.

For these reasons, context-aware research has evolved from the direct use of raw sensor data to the adoption of dedicated sensor data fusion techniques. These are used to infer higher-level context from low-level raw sensor data. The resulting higher-level context information is used to build the applications.

Adopting such a kind of approach, applications are decoupled from the acquisition and processing of sensor readings. This modularity and separation of concerns facilitates the development of context-aware systems.

Attracted by the simplification of the development and maintenance of applications, the context-awareness community has developed tools that allow to organize, model and reason about contextual information [BBH$^+$10]. Using those tools, developers can concentrate on the targeted applications, easily modeling interesting context information and reasoning about it to extract higher-level information.

From Context-aware to Situation-aware Systems

The evolution of context-aware approaches showed that, although useful, abstracting from raw sensor data is not sufficient [YDM11]. Applications should not concern themselves with basic contextual information, *e.g.*, which rooms are occupied, which appliances are being used. Rather, this information should be interpreted into a higher, domain-relevant concept, such as whether a dangerous appliance was left unattended. This higher-level concept is called a *situation*.

A situation is an “abstraction of the events occurring in the real world derived from context and hypotheses about how observed context relates to factors of interest to designers and applications” [YDM11]. It is an “external semantic interpretation” of context [YDM11].

It is an *interpretation* because a situation gives meanings to context. It is *external* because the interpretation is from the point of view of applications, instead of that of sensors. It is *semantic* because the situation gives meaning to context based on structures and relationships between different types of context.

As perceived by a context-aware system, situations represent the observable phenomena of the activity. They can be imagined as materializations of the unobservable state of affairs from the inhabitant point of view. They are the manifestation of only few clues, which hide a much more complex phenomenon, like the tip of an iceberg.

The abstraction steps (from raw data to context, from context to situations) allow representing contextual information at the level that presents the fair trade-off between hiding lower-level details and preserving relevant information. Sensor data fusion techniques allow to abstract raw sensor data to concrete pieces of contextual information; then, the resulting context information can be used to model situations, by defining correlations between contexts; finally, the recognition of ongoing situations determine the behavior of the applications. Using situations, applications are thus provided with a simple representation that abstracts from complex sensor data and from basic context.

3.1.2 Main Aspects of Situation-Awareness Techniques

The context-awareness community has developed a number of tools for modeling and reasoning about situations. These are characterized by three main aspects [YDM11]: *representation*, *specification* and *reasoning*.

Representation *Representation* deals with defining primitives that are used to construct a situation's specification. There are several possible alternative choices: an example is using logical primitives that are rich enough to capture features in complicated sensor data (*e.g.*, acceleration data), domain knowledge (*e.g.*, a spatial map), and different relationships between situations. Another choice consists in adopting a simple, generic context representation model that can incorporate any kind of information.

Specification *Specification* deals with forming a situation's specification, which can be acquired by experts or learned from training data. In manual specification, experts relate "relevant contexts to a situation, decide their different contribution weights (*i.e.*, to what degree the contexts contribute to identifying a situation), and quantify their uncertainty measurements (*i.e.*, to what degree the input sensor data validate the contexts)" [YDM11]. A context model using intuitive concepts and heuristics to define situations can greatly help this task.

Reasoning *Reasoning* deals with inferring situations from basic contexts. This is done by fusing several pieces of basic contextual information. In this meaning, we also refer to reasoning as *situation recognition*. Reasoning can also deal with discovering relationships between situations and can be used to guarantee the consistency and integrity of sensed information.

These three aspects are tightly related to each other and provide a scheme for classifying existing situation-awareness techniques. In particular, the dimension of specification divides existing techniques in two main categories: learning-based and specification-based approaches [YDM11]. The rationale behind such classification is that situations can either be manually specified or (semi-) automatically learned.

Learning-based approaches are used to properly specify situations that involve a large number of noisy sensor data. Techniques in machine learning and data mining are borrowed to explore association relations between sensor data and situations.

Specification approaches represent expert knowledge in (fuzzy) logic rules and apply reasoning engines to infer proper situations from current sensor input. We will use such classification in the forthcoming analysis of state-of-the-art techniques.

3.2 Quality of Situation-Awareness Techniques

Apart from the approach to the specification, the dimensions of representation and reasoning (*cf.* §3.1.2) also influence the usability and effectiveness of the techniques in modeling and recognizing situations. In particular, existing context-awareness techniques differ in several aspects, including the ease of use, the expressive power and the tools that they offer to deal with uncertain data. Such aspects are particularly relevant with respect to the goals of this doctoral research. This section explains why.

3.2.1 Ease of Use and Expressive Power

The ease with which developers can model situations, while being provided with highly expressive tools that can easily represent real world concepts, is important.

When designing the functionalities of a Smart Home, the interdisciplinary collaboration should be simplified by the adoption of computational models that allow a dialog between computer scientists and cognitive ergonomists, leveraging intuitive concepts and metaphors.

Given the asymmetry between the models of domestic activity produced by the cognitive ergonomists and the models of the world used by computer scientists, a common concept that facilitates the collaboration between them is necessary (*cf.* Chapter 2). **In particular, the common concept might be the situation, which is a concept considered in both the analysis of activity and the design of context-aware systems, even though from different points of view.**

3.2.2 Dealing with Uncertainty

The physical world, from which the pervasive system must capture relevant information for its functioning, presents a great deal of ambiguity and uncertainty of input. For example, the simple sensing case may report that a person is in a room when, instead, she has simply left her active badge on the desk.

This problem can be solved using more and better sensors, although often at a cost of impacting user privacy. For this reason, technical limitations should be treated as intrinsic properties of the system and their assessment should be an integrating part of the context management process, estimating the degree of uncertainty and risk connected with the inference process.

These considerations raise design issues relative to the gap between the complexity of human context and the context of the system based upon an environmental capture.

Some situations can cause the system to an inability to determine the appropriate action to take; thus, designing a context-aware system requires assessing the degree of uncertainty that affects the captured information, in order to recognize those situations and avoid unmotivated actions.

The remainder of this chapter illustrates some relevant context and situation modeling and reasoning mechanisms. We aim to describe and evaluate existing context-awareness techniques that are generic (*i.e.*, suitable for any kind of application) and are able to meet the identified quality parameters. We present and evaluate existing situation and context awareness techniques classifying them in learning-based and specification-based techniques. The next section deals with learning-based approaches, while the following one describes specification-based techniques.

3.3 Learning-based Techniques

Learning-based approaches can be classified into supervised, unsupervised and hybrid learning techniques. We first present the three categories and then evaluate their applicability to our research.

3.3.1 Supervised learning techniques

Supervised learning techniques learn the parameters of situations from training data, which have previously been labeled with the corresponding situations. Following the training, the algorithm is then able to classify unknown data.

There are a wide range of algorithms and models for supervised learning and activity recognition. These include *Hidden Markov Models* (HMM), *dynamic* and *naïve Bayesian networks*, *decision trees*, *nearest neighbor* and *support vector machines* (SVMs). Among them, HMMs and Bayes networks are the most commonly used methods in situation and activity recognition [CN09].

Figure 3.1 shows an example of an HMM for the eating activity. Based on the objects used (spoon, knife, fork or cup), which are the observable variables, we can infer the HMM states and their transitions, *i.e.*, the different phases of a meal.

Evaluation The disadvantage of supervised learning in the case of probabilistic methods is that they require a large amount of labeled training and test data. When there are a large number of situations to be identified, manual labeling of training data may represent a significant burden for developers. In addition, learning each activity in a probabilistic model for a large diversity of activities and variations in the activities, users and other variables in real world application scenarios could be deemed as being computationally and practically impracticable. The resulting models are often ad-hoc, not reusable and scalable due to the variation of the individuals' behavior and their environments [CN09]. Therefore, supervised learning techniques may have limitations in real-life deployment, where scalability, applicability, privacy and adaptability are highly concerned [YDM11].

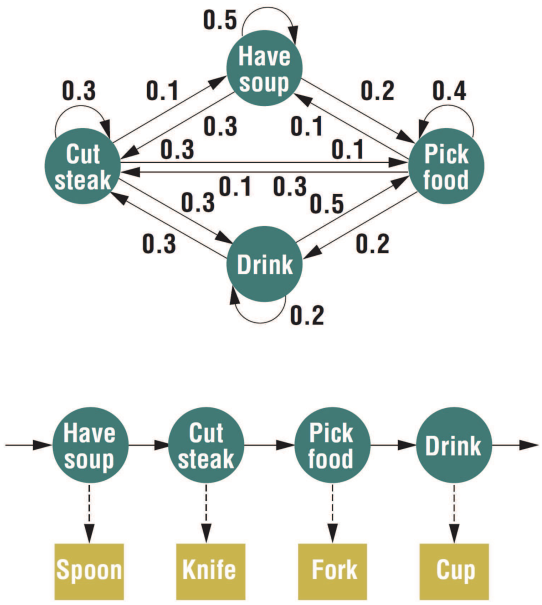


Figure 3.1: An example of a HMM for the eating activity (from [KHC10])

3.3.2 Unsupervised learning techniques

Unsupervised learning techniques directly construct recognition models from unlabeled data. These techniques start from observations and automatically discover situations by extracting features that are distinguishable from one situation to another. Such an approach employs density estimation methods or clustering techniques to discover groups of similar examples to create learning models.

Algorithms for unsupervised learning include the use of topic models [HFS08]. A number of unsupervised learning methods are also based on probabilistic reasoning such as various variants of HMMs and Bayes networks.

Evaluation Using unsupervised learning techniques to automatically discover situations can provide insights to the residents' habits and the nature of the environment [KHC10]. However, these techniques basically discover regularities and recurring patterns, which, as we previously highlighted, do not encompass the complexity of the activity (*cf.* 1.4.5). Therefore, at their current state of maturity, unsupervised learning techniques can be used to assist experts in the specification of situations but cannot suffice. They must be complemented with specification-based techniques, which allow to manually define situations that, although not discovered by unsupervised-learning techniques, are considered relevant by experts.

3.3.3 Hybrid learning techniques

Some research efforts have combined unsupervised and supervised learning in a hybrid approach to activity discovery and classification. Unsupervised learning improves the training and recognition phases of supervised algorithms by providing initial labels, finding unnoticed activities, identifying local attribute associations and finding overlapping activities [LD11]. They can also simplify the task of learning models of complex activities by limiting the labeling to individual activities, automatically mining patterns for interleaved and concurrent activities [GWT$^+$09].

Some works have even tried to map the resulting labels to real activities using automatically mined information about object use, obtained from texts available on the Internet [WPC05, GCTL10].

Evaluation The labels and trained classifiers resulting from hybrid learning techniques are able to recognize patterns and map them to real-world situations, but still cannot be used to provide adapted functionalities. Slight differences in a particular situation can ask for very different functionalities. So, these can only be designed leveraging expert knowledge about the domestic activity and the context-aware system. Designing and realizing a system that automatically discovers and learns situations from unlabeled training data, maps these to real-world situations using automatically mined data and finally automatically generates adapted functionalities does not seem to be realizable with existing techniques.

3.4 Specification-based Techniques

Given the limitations of learning-based techniques for situation awareness, we focused on specification-based approaches. Specification-based approaches typically build a situation model with *a priori* knowledge and then reason on it with input sensor data. This section will introduce the mainstream of specification-based techniques.

3.4.1 Fact-based models

Fact-based approaches constitute formal models of context that support querying and reasoning [BBH$^+$10]. One of the well-known and mature examples of fact-based models is the *Context Modelling Language* (CML) by Henricksen *et al.* [HI06, HI04, HIR02]. CML provides a graphical notation that supports the analysis and formal specification of the context-aware capabilities of applications.

Figure 3.2 shows an example model realized with CML. It models a context-aware communication application, representing concepts like users, their location and activities, as well as the communication channels and devices that they use. Ellipses represent object types, while boxes depict the role played by an object type in an association. A certainty measure is also associated to the location of an object, in the form of a probability estimate.

Evaluation One of the main strengths of CML is its support for various stages of the software engineering process. Its graphical notation supports analysis and design of the context requirements of a context-aware application; the relational representation and grammar for high-level context abstractions support runtime representation and querying.

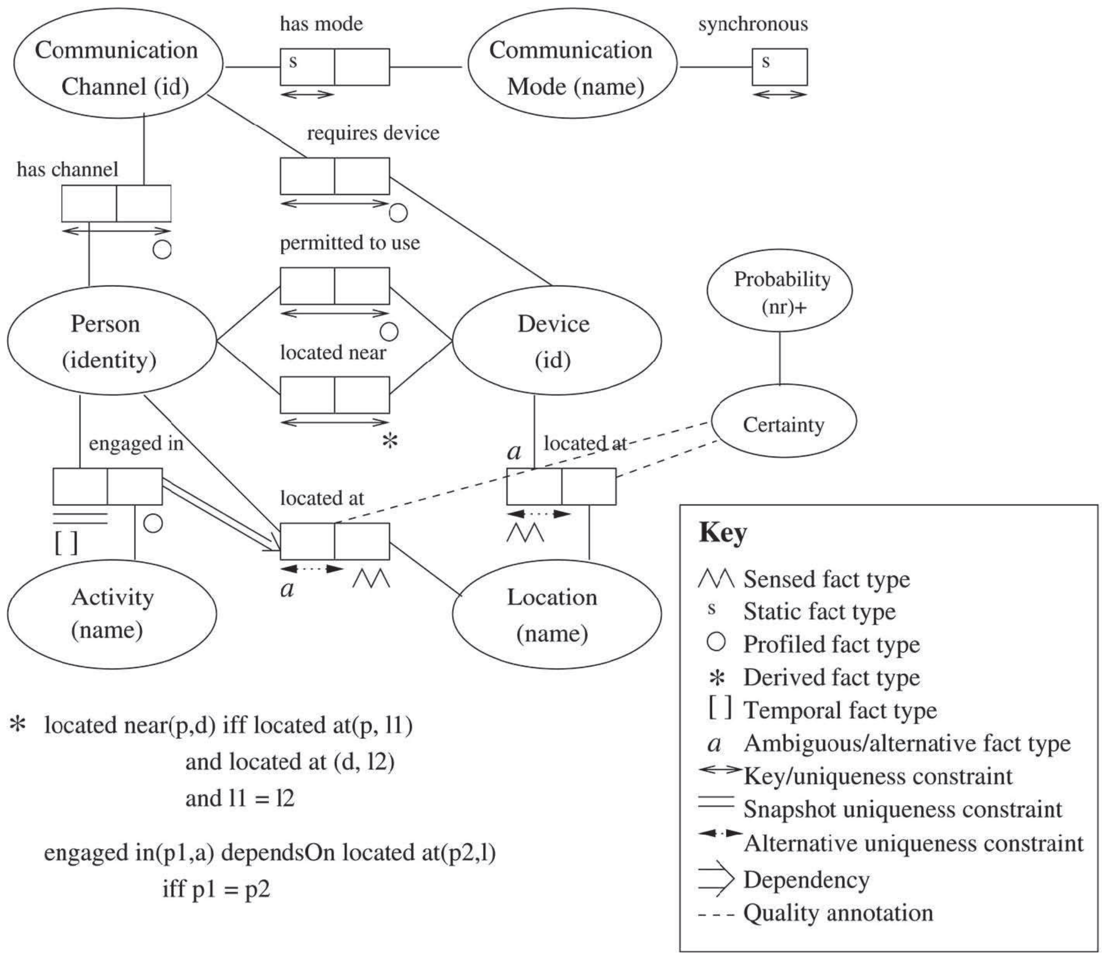


Figure 3.2: An example CML model (from [HI06])

However, CML has the weakness of adopting a "flat" information model, in that all context types are uniformly represented as atomic facts: it is not possible to represent hierarchical structures or to specify the relevance of the particular dimensions of context. This is a drawback for many context-aware applications. For instance, location may have greater importance than other types of information in a location-based application.

3.4.2 Spatial models

Space is important in many context-aware applications. Most context definitions mention space as a vital factor and space is generally well suited to organize and efficiently access context information [BBH+10]. Spatial existence also serves well as an intuitive metaphor for non-physical context information [PZL06].

The spatial context model developed in the Nexus project (called Augmented World Model) [NGS+01] is an object-based class hierarchy of context information. Almost all objects (real and virtual) are modeled with a location, either by their physical location or by a meaningful association metaphor, as shown in Figure 3.3. The Nexus context model was designed to be sharable between different context-aware applications in a potentially global scope and thus to be scalable to a high amount of context data. In the Augmented World Model, higher-level context information (*e.g.*, situations) is not managed and has to be handled by applications.

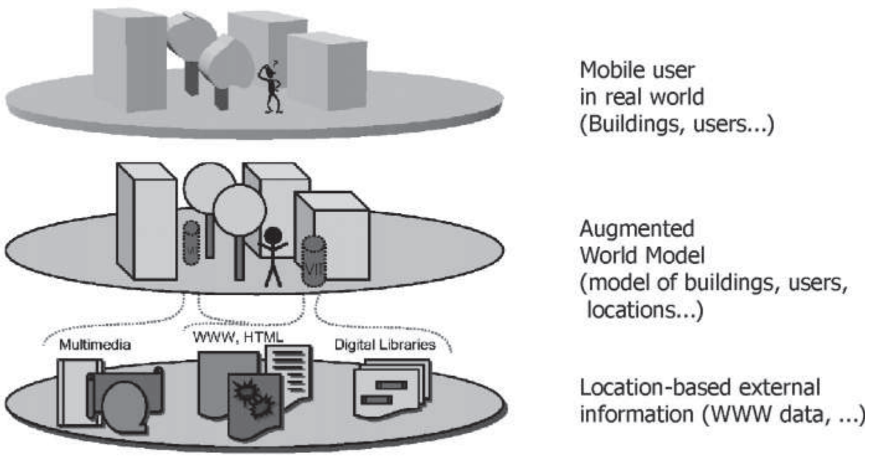


Figure 3.3: Real and Augmented Worlds in the Nexus model [NGS+01]

Even beyond the concept of spatial modeling is the *spatial computing* paradigm. Spatial computing can be viewed as a way to link spatial features in physical or virtual spaces with the specific computational actions that are taken in these points in space, on the data and active computational processes that may be located there as well as in their vicinity. In spatial computing, information acquires a new, physical meaning.

A relevant example of spatial computing is represented by *SPREAD* [CB03]. This lightweight framework for designing ubiquitous computing applications defines programming abstractions based on the properties of the physical space, like proximity between entities. Adopting the paradigm of SPREAD, application logic is directly dependent on physical properties, like physical mobility [CB03, BCPB04]. Figure 3.4 shows a hitch-hiking application that can be easily implemented using SPREAD [CB03].

Evaluation Spatial context models and paradigms allow reasoning and acting depending on the location and spatial relationships between objects. Since many context-aware applications use space as a selection criterion to retrieve other context information, it is reasonable to design context management systems to efficiently support spatial queries. In addition, if the amount of context information gets very large, it can be partitioned along the spatial dimension. In some applications, a spatial pre-selection of relevant context information could be reasonable to speed up the reasoning process by reducing the size of the knowledge base.

With respect to our research goals, a major drawback of spatial context models is the lack of support for dealing with situations. Except for what can be inferred from

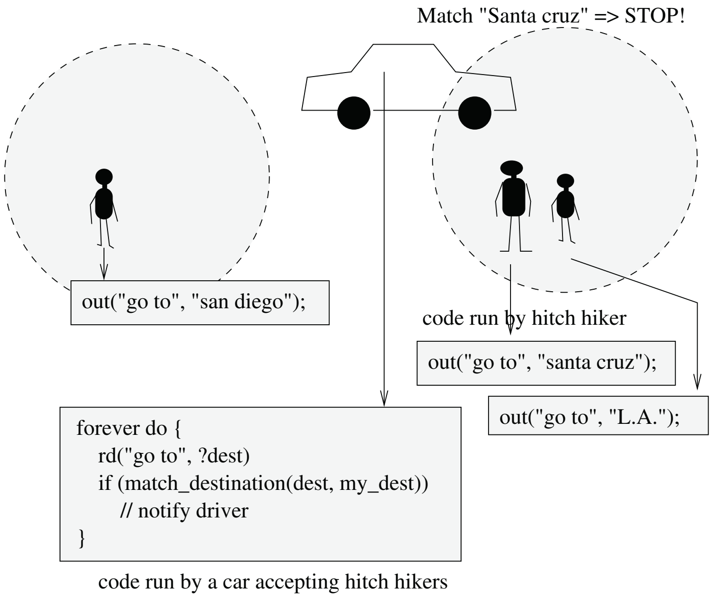


Figure 3.4: Hitch-hiker application developed using SPREAD [CB03]

spatial information, no tools are provided for modeling and reasoning about situations.

3.4.3 Logic-based models

Formal logic approaches allow to represent situations as logic specifications and to verify the integrity and consistency of those specifications. A notable example of such category of specification techniques is the *situation theory* (also called *situation calculus*) [MB97].

Situation theory is a logic language commonly used in Artificial Intelligence (AI), where a situation is modeled as the state of the real world at given moment. Based on situation theory, Akman *et al.* [AS97] defined the *extended situation theory.* In this formalism, context is the combination of facts and rules that govern the relations within the context itself.

Figure 3.5 shows a context model expressed using the extended situation theory. In the example, $\dot{a}$ is an advisor of $\dot{b}$. C is a relationship between S_1 and S_2 that indicates that if $\dot{a}$ is an advisor of $\dot{b}$, then he is usually also a committee member in $\dot{b}$'s thesis presentation (provided that condition B is satisfied).

Evaluation Formal logic approaches do not natively deal with uncertainty. Input data are assumed to be certain and the inference is also deterministically obtained as combination of conditions. To handle uncertainty, logic programming based approaches need to be combined with other techniques, like fuzzy logic, to quantify the uncertainty to be used in situation awareness.

$$S_1 = [\dot{s} \mid \dot{s} \models \ll \text{m.s.advisor}, \dot{a}, \dot{b}, 1 \gg]$$

$$S_2 = [\dot{s} \mid \dot{s} \models \ll \text{m.s. committee-member}, \dot{a}, \dot{b}, 1 \gg]$$

$$C = (S_1 \Rightarrow S_2 \mid B)$$

Figure 3.5: Context model expressed using the *extended situation theory* [AS97]

3.4.4 Fuzzy logic

Fuzzy logic is an extension of Boolean logic, from *crisp* sets to *fuzzy* sets. The theory of fuzzy sets is widely used to deal with uncertainty of vagueness. This is represented using a *membership function*, specifying to what degree an element belongs to a fuzzy set [Zad65].

In context-aware computing, fuzzy logic is especially used to map sensor data to *linguistic variables*, which correspond to social or conceptual conditions [DHKZG08]. It allows to model imprecise knowledge, *e.g.*, an approximation of a numerical value or vague information. For instance, a temperature of 10 degrees Celsius can be contained in the fuzzy set 'cold' with a fuzzy value of 0.8 and in the fuzzy set 'hot' with a fuzzy value of 0.2.

Situations are modeled as combinations of linguistic variables using fuzzy logic operations, including intersection, union, complement and modifier of fuzzy sets. Figure 3.6 illustrates the union and intersection of fuzzy sets [Zad65].

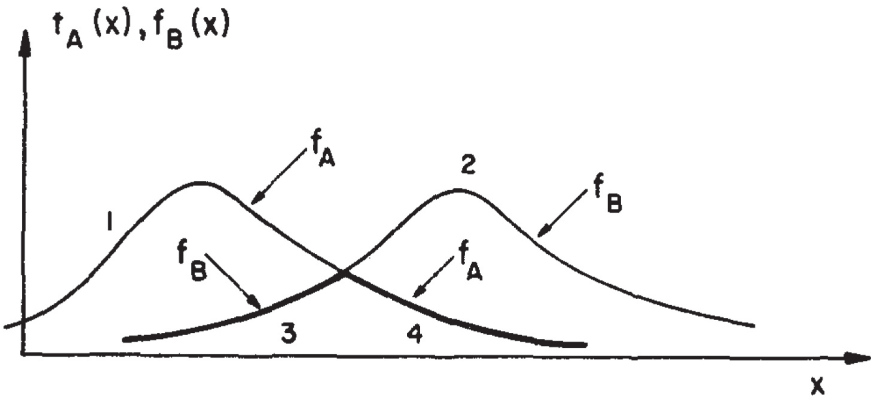


Figure 3.6: Illustration of the union and intersection of fuzzy sets (from [Zad65])

Anagnostopoulos *et al.* [ANH07] apply fuzzy inference to find a situation that is most similar to the current unknown situation by evaluating the similarity of the specifications of situations in a knowledge base and the current context input.

A fuzzy function is applied to evaluate the degree of membership in a situational involvement that refers to the degree of belief that a user is involved in an estimated situation. This situational involvement measurement will be used to help a system deciding which tasks should be executed to react to the inferred situation [ANH07].

Evaluation The drawback of using fuzzy logic is in the impossibility of modeling different degrees of importance of the context information with respect to a situation. Different contexts can contribute, through situation determination rules, to the assessment of the degree of the user's situational involvement. However, there is no way to define more complex relations between contexts and situations. Real-world situations, instead, are often the result of articulated combinations of contexts, with different weights and specific relationships.

3.4.5 Temporal logic

Temporal logic is a well-established area of AI, which has been applied to representing and reasoning on temporal features and constraints of context and situations.

Augusto *et al.* [ALM$^+$05] introduce the temporal operators *ANDlater* and *ANDsim* in Event-Condition-Action rules, upon which temporal knowledge on human activities can be specified. Rules specify situations like 'user has fainted' as combinations of ordered sequences of sensor events like: 'activation of the RFID sensor in the kitchen', then 'activation of the RFID sensor while the user is passing through the door between the kitchen and the reception area, then no detection of any movement' [ALM$^+$05].

Gottfried *et al.* [GGH06] apply qualitative AI techniques in dealing with temporal and spatial knowledge in Smart Homes. *Allen's Temporal Logic* is used to describe, constrain and reason on temporal sequences between two events, as shown in Figure 3.7. For example, given the following events: the door to the room is opened before the person is in the room and the sensor is triggered during the person's presence in the room, then the composition of these events suggests that the door is opened before the sensor is triggered.

Evaluation Temporal logic allows modeling temporal relationships between events in an easy and intuitive way. Furthermore, existing works also address the management of uncertainty in temporal logic [ALM$^+$05]. However, assuming that activities and manifestations of human context always follow predefined orders can easily bring to wrong conclusions. Human activity is complex and articulated, as well as variable in its execution, so adopting temporal constraints as the core description concept does not seem to facilitate the task of modeling real-world situations.

3.4.6 Ontologies

Ontologies provide modeling primitives to define classes, individuals, attribute properties, and object properties (*i.e.*, relations between objects). For example, the *is-a* property is one of the most useful properties in modeling the abstraction level of the domain concept: 'Dining room' is-a 'Eating Activity Space' [NKG$^+$15].

Ontologies provide a formal way to represent sensor data, context and situations into well-structured terminology, which makes them understandable, sharable and reusable by both humans and machines.

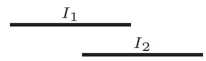
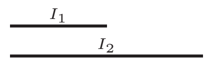
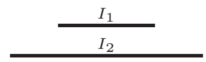
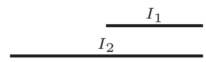
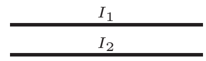
Relation	*Illustration*	*Interpretation*
$I_1 < I_2$ $I_2 > I_1$		I_1 before I_2 I_2 after I_1
$I_1 \mathsf{m} I_2$ $I_2 \mathsf{mi} I_1$		I_1 meets I_2 I_2 met by I_1
$I_1 \mathsf{o} I_2$ $I_2 \mathsf{oi} I_1$		I_1 overlaps I_2 I_2 overlapped by I_1
$I_1 \mathsf{s} I_2$ $I_2 \mathsf{si} I_1$		I_1 starts I_2 I_2 started by I_1
$I_1 \mathsf{d} I_2$ $I_2 \mathsf{di} I_1$		I_1 during I_2 I_2 contains I_1
$I_1 \mathsf{f} I_2$ $I_2 \mathsf{fi} I_1$		I_1 finishes I_2 I_2 finished by I_1
$I_1 = I_2$		I_1 equals I_2

Figure 3.7: Allen's thirteen temporal relations, used by Gottfried *et al.* [GGH06]

Ontologies can be used to support reasoning. The inference is well supported by mature algorithms and rule engines. Such reasoners can be used to infer new facts from sensor data (*e.g.*, the situation 'sleeping' from the location and posture of a person [GPZ05]) and to discover inconsistencies in sensed context (*e.g.*, the same person is detected in two different rooms at the same time).

Another possible use of ontologies in activity recognition is to validate the result inferred from statistical techniques [RB09]. Ontologies are used to model the relationships between the domain concepts including rooms, objects, and activities, as shown in Figure 3.8. In a situation where the sensors report the user's current location as 'LivingRoom' and a statistical technique infers the current possible activities as 'BrushTeeth' and 'Reading', the ontological reasoner will filter out 'BrushTeeth' and infer 'Reading' (see Figure 3.8).

Evaluation Using ontologies, activities are supposed to follow strict rules and constraints. In reality, a person might start brushing his or her teeth in the restroom, but then continue it while moving to the living room to do something else concurrently.

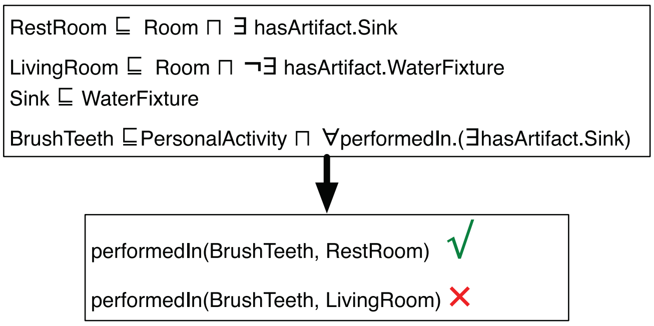


Figure 3.8: An example of the use of ontologies in deriving new facts [YDM11]

Furthermore, labeling rooms with the activities that can and cannot be performed there [NKG+15] is not appropriate. Analysis of domestic activity shows that the activity is not specific to any place. Any space is potentially an 'eating space' and the same activity can potentially be performed anywhere. For example, a person might decide that he or she wants to eat a sandwich in front of the TV, lying on the sofa.

Ontologies and the associated inference tools focus on deterministically deriving new facts from structured premises. However, human activity and situations are rather characterized by the absence of rules than by their enforcement. Uncertainty intrinsically characterizes the considered scenarios. For this reason, structured modeling and deductive reasoning fail to provide suitable tools for modeling and reasoning about real-world activity and situations.

3.4.7 Belief functions theory

The *Belief Functions Theory* (BFT), also known as *Dempster-Shafer theory* [Sha76], is a mathematical theory of evidence. It can be used to combine evidence from differently reliable sources and propagate uncertainty from the different pieces of evidence up to situations, consequently providing an indication of the certainty of inferences. As a generalized probability approach, BFT quantifies and preserves ignorance due to the lack of information.

BFT has been applied to situation recognition in context aware systems. A notable example is the work from McKeever *et al.*, who applied the BFT to incorporate sensor uncertainty into sensor evidence and to fuse this evidence to infer situation occurrence. This is done with minimal use of training data [MYCD09].

To propagate evidence across layers of context, the authors use a multi-layered hierarchy consisting of sensors, abstracted context and situations [MYCD08], as illustrated below (*cf.* Figure 3.9):

1. sensor readings are abstracted to more human-understandable *context values*;

2. evidence from each context value is transferred to one or more situations, indicating that a particular context value is evidence of the situation occurrence;

3. evidence from different context values for the occurrence of the same situation is combined using a combination rule;

4. higher-level situations may also be inferred from lower-level situations.

At each point in time, the situation with the greatest belief (evidential support) is believed to be occurring.

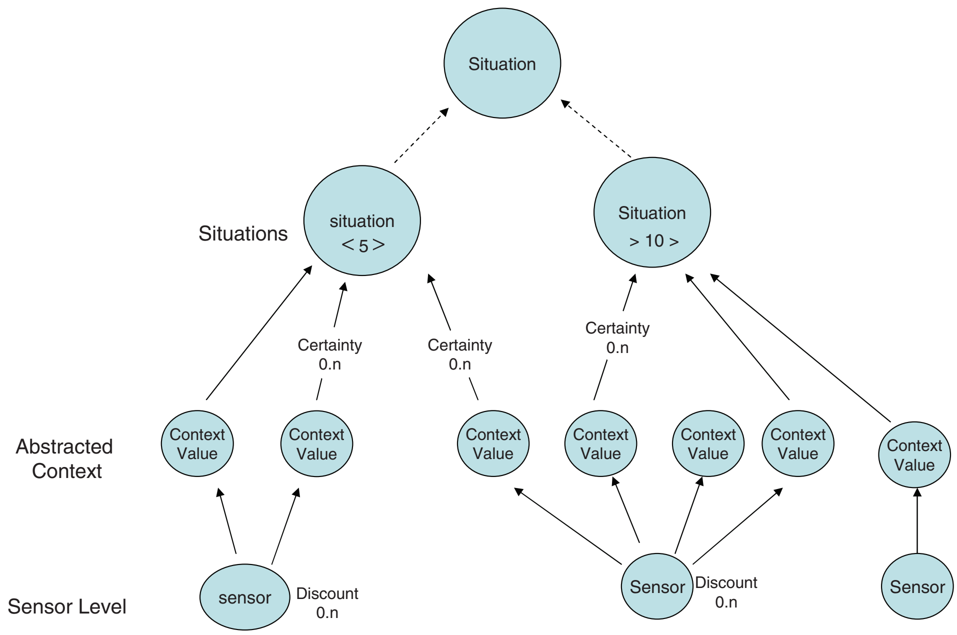


Figure 3.9: Situation inference using the *belief functions theory* (from [MYCD09])

Evaluation Using BFT as a specification-based technique in situation awareness research has the main advantage of being capable of representing multiple types of uncertainty. Imprecision, ignorance, inaccuracy and unreliability of sources can all be taken into account using a unified modeling and reasoning framework.

However, existing approaches using BFT for situation inference only use binary sensors (the sensor is either triggered or not triggered) [HNM+09, MYCD09]. In real-world scenarios it might be useful to use sensors returning continuous values (vibration, sound level, *etc.*).

Furthermore, existing approaches make a *closed-world* assumption. That is, a situation is ongoing at a given moment if the belief of its occurrence is the highest compared to the other situations [MYCD09]. This means that all the situations that may possibly occur in the domestic environment have to be modeled, which is clearly not possible.

Finally, existing approaches have only dealt with mutually exclusive situations [MYCD09]. This means that the case of multiple situations occurring at the same time is not considered.

3.4.8 Complex Event Processing (CEP)

Complex Event Processing (CEP) [EB09] encompasses methods, techniques and tools for processing events while they occur, *i.e.*, in a continuous and timely fashion.

An event is an actual occurrence or happening that is significant (*i.e.*, it falls within a domain of interest of the system), instantaneous (*i.e.*, it takes place at a specific point in time) and atomic (*i.e.*, it either occurs or not) [WGET08].

CEP was used in database management systems to allow issuing persistent queries, which are processed as events occur. Traditional queries, instead, are executed on a static database. This is graphically illustrated by Figure 3.10.

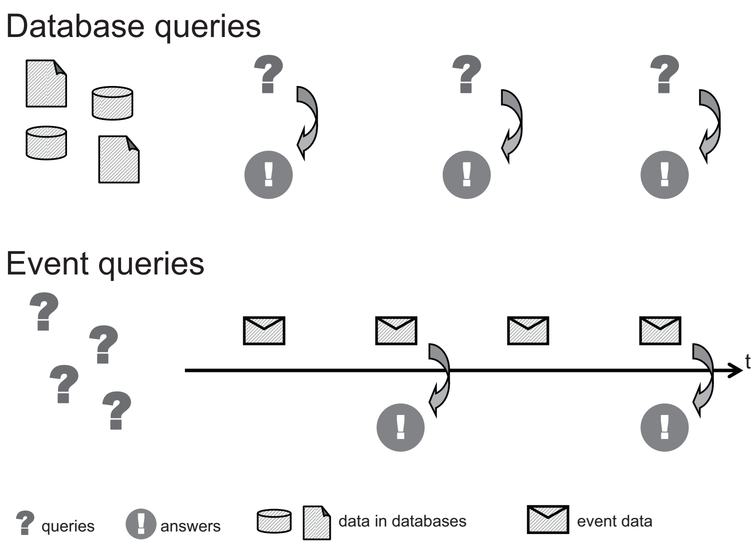


Figure 3.10: Difference between database and event queries (from [EB09])

CEP derives valuable higher-level knowledge from lower-level events; this knowledge takes the form of so-called *complex events*, that is, situations that can only be recognized as a combination of several events.

Wasserkrug *et al.* [WGET08] present an efficient mechanism for complex event processing (or, in their vocabulary, event *materialization*) under uncertainty. Given a set of events and a set of rules, this mechanism materializes new events, their associated data and their level of uncertainty.

In this system, events are recorded using an Event Instance Data (EID) relation, a tuple that captures the information of the system. For instance, the structure of an EID for daily sales is *DS(EID, Date, dailySales)*, representing the dates and the daily sales of a particular product [WGET08]. When an event is uncertain, it is represented by several tuples, where each such tuple has an associated probability.

Rules define the necessary conditions for the materialization of new events. Rules are composed of five elements:

1. *selection function*, filtering events relevant to materialization, called *selectable* events;
2. *predicate*, determining when events become candidates for materialization;
3. *association* function, defining how many events should be materialized, as well as which subsets of selectable events are associated with each materialized event;
4. *mapping* function, determining the attribute values of the materialized events;
5. *probability* function, determining the probability of the materialized events.

To summarize, the first-class concept is the event, which, combined with previously triggered events, is abstracted (or "materialized") through a rule to a complex event.

Evaluation To the best of our knowledge, existing CEP mechanisms, including that proposed by Wasserkrug *et al.*, make a closed-world assumption with regard to event occurrence: an event has not occurred unless it is explicitly captured as an EID.

In the context of our research, sensors and other providers of context information are not totally reliable. The information they provide is imprecise, uncertain and some pieces of information can also be missing. For instance, a person entering a room might not be detected by the system, because of sensor failure or noise. Using the language described by Wasserkrug *et al.*, defining one of such situations requires defining several subsituations. Each of those models a possible combination of events, which corresponds to having failed to observe one, two, three, *etc.* particular events.

This characteristic alone makes unpractical the use of CEP as the context model for our research. The context management mechanism that we target must place the situations at the core of the model and provide simple and intuitive tools to represent real-world situations. This includes the ability to specify *fuzzy* aspects of situations and other consequences of uncertainty, like the failure in observing an event.

3.5 Overall Evaluation of Context-Awareness techniques

We illustrated the evolution of the approaches in the context-awareness research domain and presented existing techniques, classifying them in learning-based and specification-based approaches.

In this section, we provide an overall evaluation of the illustrated techniques, highlighting the need for a unifying theory that fits the requirements of this doctoral research. This will allow us to introduce the *Context Spaces Theory*, which was used to build the contribution of this doctoral research (*cf.* §3.6).

3.5.1 Learning-based Techniques: Not Suited to This Research

Section 3.3 presented some situation recognition algorithms that are based on probabilistic learning models. A major strength of such techniques is that they are capable of handling noisy, uncertain and incomplete sensor data. Probabilities can be used to model uncertainty and also to capture domain heuristics, *e.g.*, to specify that some situations are more likely than others.

However, when applied to situation recognition in the context of the domestic environment, these techniques show important drawbacks, due to their reliance on regularities and recurring patterns. We previously highlighted that such techniques cannot encompass the complexity of the activity (*cf.* §1.4.5).

Furthermore, learning-based techniques typically require extensive training data from the target environment in order to develop the learning model. Reasoning is hidden from the user and developers to varying degrees, depending upon the learning technique, thus challenging the user's and developers' ability to scrutinize decisions [YDM11].

In addition, it should be noticed that the performance of probabilistic graphical models and machine learning algorithms strongly depends on the instrumentation choices, thus limiting their scope. For instance, they can be used with interesting results if the instrumentation includes wearable sensors, video cameras and electronic tags applied on everyday objects [WOC+07].

When, instead, the instrumentation is lighter, for instance only including environmental sensors, such techniques become difficult to apply. One of the constraints of this doctoral research, directly following from the goal of realizing an acceptable Smart Home for final users, is to use lightweight instrumentation.

For this reason, the contextual abstractions that can be obtained are very simple, like the presence or the agitation of someone in a room. Inferring the precise activity of inhabitant(s) becomes very difficult. **These considerations led us to concentrate on specification-based techniques as more interesting candidates for solving our research challenges.**

3.5.2 Specification-based techniques: Need for a Unifying Theory

We highlighted the drawbacks of each particular class of specification-based techniques in Section 3.4. Some techniques lack suitable support for reasoning about situations and high-level contextual abstractions. Others lack support for considering and reasoning about uncertain context information and focus on deriving new information from observed facts, using fixed schemes and rules. Some techniques use flat information models, making impossible the specification of different degrees of relevance or interdependency of contextual information with respect to situations. Finally, some techniques rely on the closed-world assumption, requiring complicated modeling and resulting in ineffective management of incomplete information.

To summarize, we can extrapolate two fundamental properties that an effective context modeling and reasoning technique should offer [DHKZG08, YDM11]:

1. Given the imprecision of sensor data, the satisfaction of a condition in a situation model should not be a crispy Boolean value (either true or false) but should instead take into account the uncertainty of the input.

2. In real-world situations, each condition does not contribute to the situation to the same degree. It should be possible to weigh the relevance of each piece of

evidence, to aggregate several pieces of evidence and to make a decision despite conflicting evidence.

Fuzzy logic, with its strength in dealing with imprecision, has been applied to solve the first issue (*cf.* §3.4.4), but not to tackle the second one. Evidence theories like Dempster-Shafer theory have been used to solve the second problem, but with several limitations (*cf.* §3.4.7).

The identified features, lacking in the techniques illustrated above, have to be provided by a unified technique for context and situation modeling and reasoning. Such a technique, called the *Context Spaces Theory*, is illustrated in the next section. One of the contributions of this doctoral research is in the adaptation of such a theory to build a multi-layered context-aware system (*cf.* Chapter 4) .

3.6 Context Spaces Theory

This section describes the Context Spaces Theory, which constitutes the set of modeling and reasoning tools that we decided to adopt. It is a unifying theory that solves the issues highlighted in the previous section.

The Context Spaces theory was introduced by Amir Padovitz as part of his doctoral work at Monash University, Australia [Pad06]. The purpose of the theory is to provide the tools to manipulate, use and reason about context.

This section presents an overview of the theory, followed by a description of the basic modeling concepts and of the tools for reasoning on the occurrence of situations. Then, the available tools for handling uncertainty and fuzzy/partial knowledge are introduced. The following Section 3.7 is dedicated to the evaluation of the Context Spaces theory with respect to this doctoral research goals and requirements.

3.6.1 Overview of the Theory

The Context Spaces modeling relies on a structure of context management organized in three levels: sensors, context and situations (see Figure 3.11).

Starting from data captured by sensors, the theory offers methods and algorithms to interpret and process these data and arrive at a representation of the *context*, including facts, assumptions and predictions. Reasoning mechanisms are then applied on top of this context representation in order to produce an answer (and a degree of confidence in the answer) to the question “is a given *situation* currently occurring?”.

We now illustrate the abstractions and modeling tools offered by the Context Spaces theory, as well as the context reasoning mechanisms.

3.6.2 Context Spaces Model

The Context Spaces model uses geometrical metaphors to describe context and situations. It relies on the following basic abstractions and modeling tools: the *Application Spaces*, the *Context Attributes*, the *Situation Spaces* and the *Context States*.

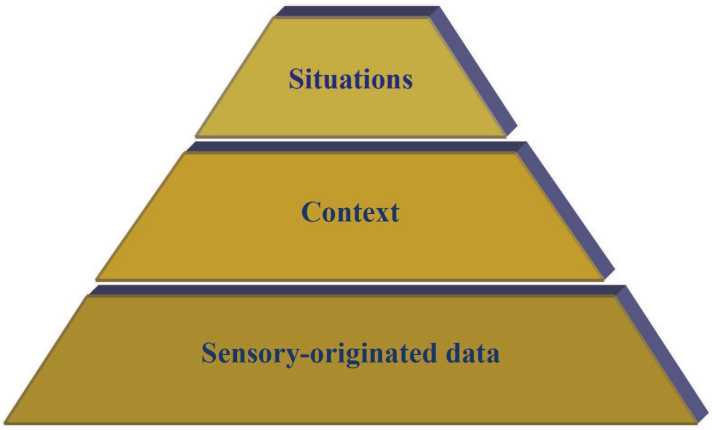


Figure 3.11: Context-Situation Pyramid (from [Pad06])

Context Attribute A *Context Attribute* represents a basic component of the context model, the entry point of the modeling and capture of context. It is an attribute whose value depends on the state of the real world. The Context Attribute may represent basic information, such as the value returned by a physical sensor (*e.g.*, the temperature in a room), or more complex concepts, returned by virtual sensors (*e.g.*, the state of an appliance). Context Attributes are denoted by a^i. The value of a sensor reading at time t is the *Context Attribute value* at time t (denoted by a^i_t).

Figure 3.12 shows a screenshot of a context-aware application exploiting the Context Spaces theory. On the left of the figure, the column *Sensor Types* shows some examples of Context Attributes.

Using a geometrical metaphor, the Context Attributes are dimensions of a multi-dimensional space, as depicted in Figure 3.13. In addition to providing the entry point for the capture of context, Context Attributes are also used to model the *Situation Spaces*.

Situation Space A *Situation Space* models a real-life situation. It allows to model a relevant situation so as to recognize it when it occurs. Situation Spaces are defined based on the Context Attributes.

A *Situation Space* is a tuple of regions of acceptable values of Context Attributes, denoted by $S_j = \left(A^j_1, A^j_2, \ldots, A^j_n\right)$. An acceptable region A^j_i is defined as a set of elements V that satisfies a predicate P, *i.e.*, $A^j_i = \{V \,|P\,(V)\,\}$.

For instance, the Situation Space *Ironing* can be modeled combining the information that a person is present in a room with the fact that the iron is on and that it is being moved. The Situation Space *Ironing* is thus a geometric shape, where the dimensions are the Context Attributes: *Presence*, *Iron status* and *Iron movement*, respectively (see Figure 3.13).

The right-hand side of Figure 3.12 shows examples of Situation Spaces corresponding to real situations, while Figure 3.13 depicts Situation Spaces as multidimensional

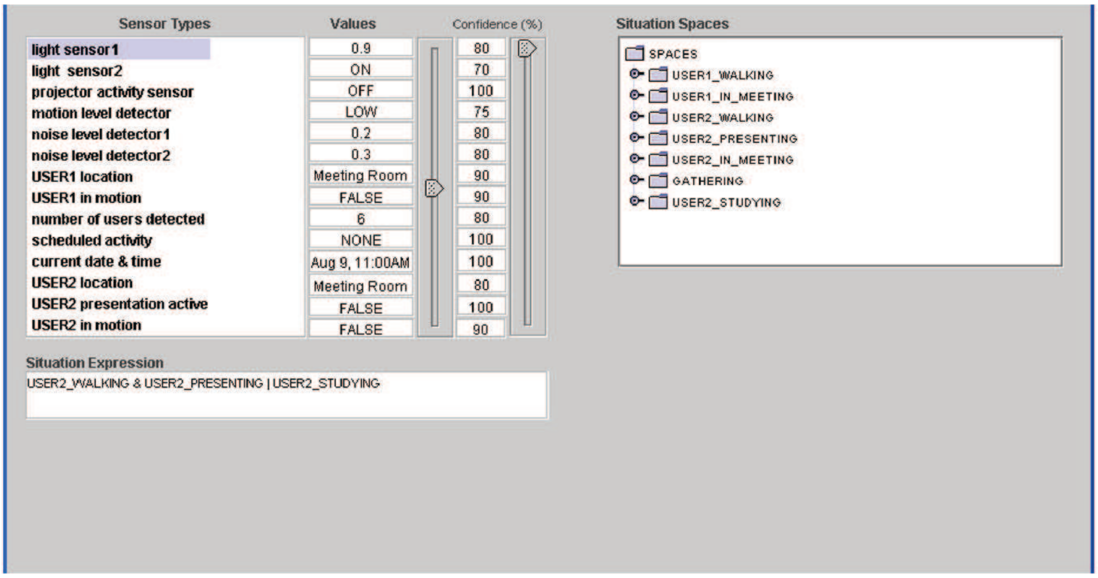


Figure 3.12: The main concepts of the Context Spaces Theory (from [Pad06])

solids (hypersolids).

Context State The *Context State* is a collection of Context Attribute values at time t and is denoted as $C^t = \left(a_1^t, a_2^t, \ldots, a_n^t\right)$, where n is the number of considered Context Attributes. Directly following from its definition, the Context State is represented as a point in the multi-dimensional space. When time flows, the Context State changes and its trace draws a trajectory in the space, as depicted in Figure 3.13. The Context State at a given moment is analyzed to infer the ongoing Situation Spaces.

Application Space The *Application Space* is the universe of discourse in terms of possible information (which is sensed/discovered/computed) for an application. It is defined by a tuple whose members represent all available Context Attributes and possible values for each Context Attribute. It is represented with the metaphor of a multidimensional space where the dimensions are represented by the Context Attributes. It is within this space that Situation Spaces are localized, shown as hypersolids in Figure 3.13. Several Application Spaces can coexist, each consisting of certain dimensions (Context Attributes) and containing some Situation Spaces.

3.6.3 Context Spaces Heuristics

To extend the modeling principles presented above, the Context Spaces *heuristics* allow modeling other aspects of real-world situations as Situation Spaces. We now present some of the available heuristics, giving examples that clarify their use.

Relevance A specific *relevance* is assigned to each Context Attribute with respect to a Situation Space. Given a situation space $S = (A_1, A_2, \ldots, A_n)$, a relevance function

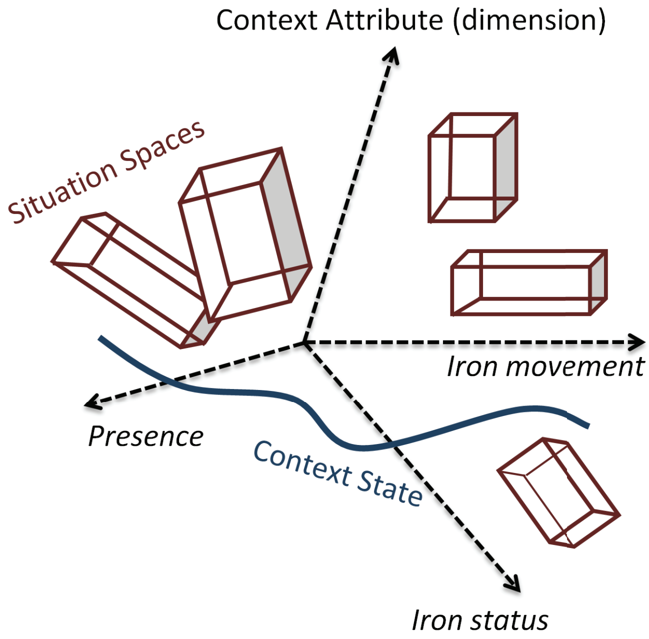


Figure 3.13: The Context Spaces model, a multidimensional space metaphor

associates weights $w_1, w_2, \ldots, w_n$ with regions of values $A_1, A_2, \ldots, A_n$ of S, respectively, where $\sum_{i=1}^{n} w_i = 1$. Thus, the relevance models the relative weight of a Context Attribute compared to the other Context Attributes describing the situation [Pad06]. For instance, in the Situation Space *Ironing*, the relevance of the Context Attribute *Presence* can be set to equal to the relevance of *Iron status* (*e.g.*, 0.4), as those two factors are equally important in the recognition of the situation. The relevance of *Iron movement*, instead, is lower than the other two (*e.g.*, 0.2), as at some point the person might leave the iron while folding clothes.

Contribution Function Another heuristic is the *contribution function*, which specifies the impact of each particular value of a Context Attribute on the recognition of the situation. Given an acceptable region of values A_i in a situation space S, corresponding to some context attribute a_i, a function κ_i assigns a contribution level $c \in [0, 1]$ to each element in the region A_i (*i.e.*, to each value of the context attribute). For instance, in the Situation Space *Ironing*, the contribution of *Presence* depends on the value of *Presence* itself: if somebody is present ($Presence = \{yes\}$), the contribution is high (*e.g.*, 1), while if nobody is detected, the contribution is low (*e.g.*, 0). In this way, we consider the actual value of each Context Attribute at the time of execution.

The contribution function must not be confused with the relevance: the relevance expresses the importance of a Context Attribute with respect to another Context Attribute for the detection of the situation, regardless of the value that the Context Attribute assumes at runtime. The contribution, instead, defines the importance of the *value* assumed by the Context Attribute.

Examples of contribution functions are presented in Figure 3.14. In the figure, *(A)* is a very simple contribution function, where every possible value, contained in the continuous acceptable region, has the same contribution. *(B)* is contribution function where the contribution is initially zero, then grows until a maximum value and remains stable, until a point where it start decreasing, finally reaching zero again. The remainder of the picture shows a contribution function made of steps *(C)* and a Gaussian-like contribution function *(D)*.

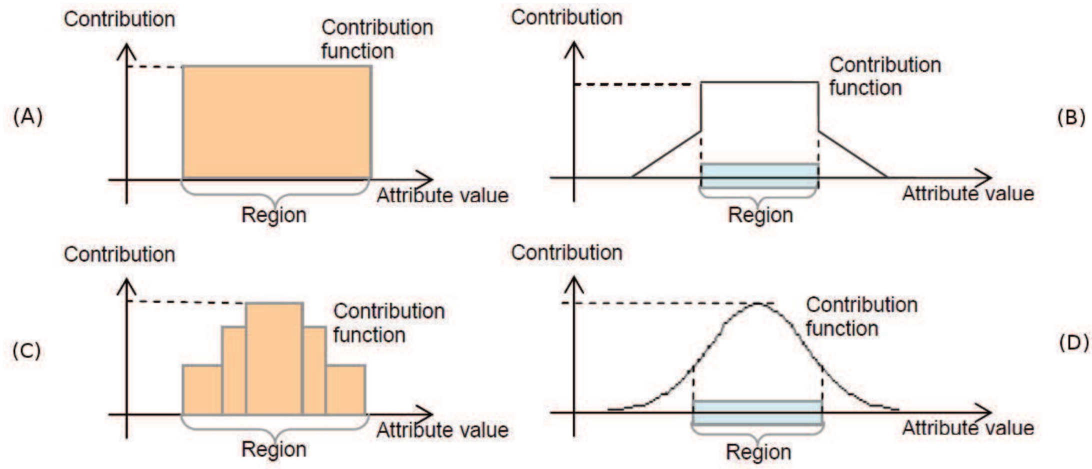


Figure 3.14: Examples of contribution functions (from [Pad06])

Additional heuristics The original Context Spaces theory and its extensions define additional heuristics and modeling tools. These include ways of:

- specifying the contribution of the fact that all the Context Attributes' values are contained in the respective acceptable regions [Pad06];
- defining *asymmetrically contributing* Context Attributes [Pad06], *i.e.*, Context Attributes that are relevant for the situation only if their value falls within the corresponding acceptable region; otherwise, they are considered as missing Context Attributes (cf. §3.6.7);
- modeling the contribution of complex combinations of Context Attribute regions [BZ11a].

We will not present the details of such heuristics for the sake of brevity.

3.6.4 Reasoning and Decision Making

Reasoning

The *reasoning* process consists in calculating, through a *matching function* μ, the *degree of confidence* in the occurrence of a situation. The matching function takes a context state at time t, a situation space S, a set of functions $\sum$ (the heuristics) and returns a number between 0 and 1, *i.e.*, $\mu\left(C^t, S, \sum\right) \in [0,1]$.

Using the heuristics described above, the confidence in a situation space S at time t is calculated as:

$$\mu\left(C^t, S, \sum\right) = \sum_{i=1}^{n} w_i \kappa_i(a_i^t), \tag{3.6.1}$$

where w_i is the weight of Context Attribute i for the situation and $\kappa_i(a_i^t)$ is the value of the contribution function for the Context Attribute i when applied to the current value a_i^t.

Reasoning on Complex Situation Expressions

The Context Spaces theory allows reasoning about the occurrence of complex situations, obtained as logical expressions composed of elementary Situation Spaces. For this, the theory provides tools to combine together several Situation Spaces and calculate the value of confidence on the resulting complex situation.

The illustration of the algorithms allowing the combination of situations goes beyond the objectives of this document. We can put forward that a system using the Context Spaces theory may reason about the co-occurrence of several situations and even compare between two complex expressions to determine which combination of situations is most likely to be occurring.

Decision Making

The decision about the occurrence of a situation S is calculated by a Boolean *inference function* γ:

$$\gamma = \left(\mu(C^t, S, \sum) \geq \varepsilon\right), \tag{3.6.2}$$

where $\varepsilon \in [0,1]$ is a confidence *threshold*. In other words, a situation is considered to be ongoing if its confidence, calculated using the heuristics, exceeds a fixed threshold.

We presented the basic modeling, reasoning and decision-making tools of the Context Spaces theory. We now proceed by introducing some important aspects of the theory: the tools to handle uncertainty, fuzziness and ignorance.

3.6.5 Handling Imperfections of Sensors

The Context Spaces theory incorporates the impact of sensor inaccuracies and unreliability in the calculation of the contribution level of a Context Attribute at runtime. This is done by integrating knowledge about sensor imperfections as part of the reasoning process. The basic idea behind this heuristic is that the greater the likelihood of the value of a Context Attribute being contained in a region, the greater the contribution assigned to that Context Attribute and *vice-versa*.

The heuristic replaces the basic contribution function in the calculation of the confidence, as follows:

$$\mu\left(C^t, S, \sum\right) = \sum_{i=1}^{n} w_i Pr\left(\hat{a}_i^t \in A_i\right), \tag{3.6.3}$$

where $Pr\left(\hat{a}_i^t \in A_i\right)$ represents the probability of a Context Attribute correct value (denoted by $\hat{a}_i^t$) being contained within the region A_i. In other words, $\hat{a}_i^t$ represents the correct value (or true state) of the phenomenon and the sensed (and possibly imperfect) value is denoted by a_i^t.

The Context Spaces Theory defines two different methods to compute the probability $Pr\left(\hat{a}_i^t \in A_i\right)$ at runtime, as illustrated below.

Estimating the Reliability of Sensors

The first method incorporates sensor *reliability* in the calculation. Reliability specifies how dependable and consistent the sensor is [Pad06]. For instance, if a sensor is known to be mostly accurate but faulty 10% of the time, the confidence in a reading being contained in the acceptable region is reduced consequently, *i.e.*, $Pr\left(\hat{a}_i^t \in A_i\right) = 0.9$.

Estimating the Accuracy of Sensors

The second method offered by the Context Spaces theory to compute $Pr\left(\hat{a}_i^t \in A_i\right)$ consists in evaluating the impact of the inherent *inaccuracy* of sensors [Pad06]. Inaccuracy indicates the distance of measurement results from the true value of the phenomenon. If sensor reading errors or their magnitude can be characterized, then such information can contribute to the estimation of the probability as follows:

$$Pr\left(\hat{a}_i^t \in A_i\right) = Pr\left(e_i \text{ between } a_i^t - min(A_i) \text{ and } a_i^t - max(A_i)\right), \qquad (3.6.4)$$

where $e_i = a_i^t - \hat{a}_i^t$ denotes the inaccuracy of the sensor reading and $min(A_i)$ and $max(A_i)$ denote the minimum and maximum values of an acceptable region of values A_i (corresponding to the Context Attribute a_i), respectively. The practical implication of this is that, given a sensor reading and an estimation of the reading error distribution, we can estimate the probability that the correct value is in the acceptable region [Pad06]. Figure 3.15 illustrates three different cases, where the probability of containment is high (case A), lower (case B) and very low (case C), depending on the size of the region and on the inaccuracy of sensors.

3.6.6 Handling Uncertainty and Fuzziness

The previously illustrated heuristic for handling sensor imperfections does not deal with fuzzy aspects of situations. That is, Equation 3.6.3 deals with the probability that a Context Attribute value is *contained* in an acceptable region, regardless of where exactly inside the region the value may fall.

To tackle this issue, Delir Haghighi *et al.* have integrated fuzzy logic principles into the Context Spaces Theory, proposing the *Fuzzy Situation Inference* (FSI) technique [DHKZG08]. Among other things, FSI is able to reflect delta changes of context in the situation inference results [DHKZG08].

To this end, the FSI model maps situation modeling concepts and reasoning methods of the Context Spaces model into a fuzzy structure and tailors them to conform to fuzzy logic principles:

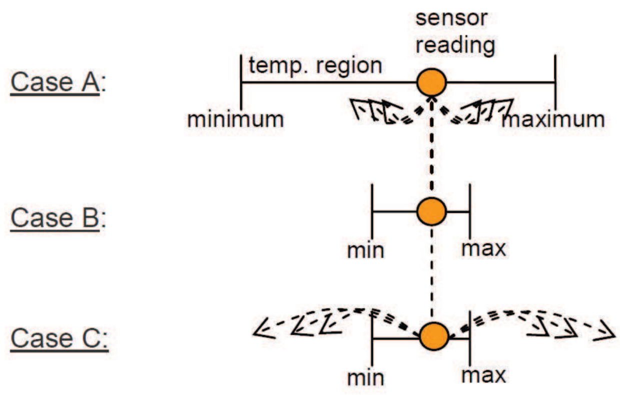


Figure 3.15: The probability of containment (from [Pad06])

1. The concept of *acceptable region* of values of a Context Attribute for a Situation Space is replaced by a *fuzzy set*. While for crisp sets an item is either contained or not contained in the set, in a fuzzy set an item has a *membership degree* between 0 and 1.

2. Context Attributes are replaced by *linguistic variables*. Values that linguistic variables take, called terms or *fuzzy variables*, are couples of the kind *(value, membership degree)*.

3. A Situation Space is defined by a set of fuzzy sets that are expressed as a FSI *rule*, which consists of multiple conditions joined with the *AND* operator, where each condition can itself be a disjunction of conditions (*i.e.*, using the *OR* operator).

With FSI, the situation reasoning is thus performed with the following equation:

$$\mu\left(C^t, S, \sum, E\right) = \sum_{i=1}^{n} w_i \varphi_i \left(f(a_i^t, e_i^t)\right), \tag{3.6.5}$$

where $\varphi_i \left(f(a_i^t, e_i^t)\right)$ is the membership degree of the context attribute value a_i^t after that it has been corrected using the function $f(a_i^t, e_i^t)$. This function calculates the correct value of the Context Attribute based on the error value e_i^t. If e_i^t is a reliability rate, a_i^t is multiplied by it. Conversely, if e_i^t is an inaccuracy measure, it is added to a_i^t [DHKZG08].

3.6.7 Handling Partial Knowledge

In a context-aware system, relying on imperfect and faulty sensors, information can often be partially missing.

The Context Spaces Theory includes an algebra that allows to modify Situation Space models at runtime, so as to adapt them to the available (or unavailable) information [Pad06].

The illustration of such algebra goes beyond the scope of this dissertation. However, we will present two of the provided tools, because they are required to understand the contribution of this doctoral research.

Dissimulating ignorance A particularly useful tool that is provided by the theory is the *relative weight transformation* (or *weight recalculation*) process [Pad06], which allows to dissimulate ignorance. It consists in proportionally redistributing the original weights $w_1, w_2, \ldots, w_n$ of the Context Attributes involved in the situation in order to take into account the fact that one Context Attribute is not available.

Considering the case in which the n-th Context Attribute is missing, the algorithm of relative weight transformation calculates the new weights as:

$$\hat{w}_i = \frac{w_i}{\sum_{j=1}^{n-1} w_j}, \tag{3.6.6}$$

where $\hat{w}_i$ is the new weight of the i-th Context Attribute.

The result is that the inference on the occurrence of that particular situation only relies on the remaining Attributes' values, just like if the Attribute affected by total ignorance had not been modeled in the situation.

Considering ignorance In some cases, recalculating weights might not be desirable. If the value of a very important Context Attribute is not available, we may want to consider that as a negatively impacting factor. For instance, if the system does not know if the iron is on, it will obviously be unable to decide if *ironing* is ongoing. Recalculating weights, instead, would cause the decision to be taken as if the iron state did not exist as a factor to consider. This would obviously be inappropriate.

To model that aspect, the Context Spaces Theory implementations (*e.g.*, ECSTRA — see Section 5.2) allow to specify which Context Attributes ask for the recalculation of weights in case they are missing.

3.7 Advantages of the Context Spaces Theory

To the best of our knowledge, the Context Spaces Theory (CST) is the context-awareness technique that best fits the requirements and goals of this doctoral research. This section explains why.

3.7.1 Modeling and Reasoning Capabilities

As we showed, the CST allows to model context and situations with intuitive concepts and metaphors (*cf.* §3.6.2). It allows the specification of a large spectrum of imprecise and fuzzy aspects of situations with a hierarchical structure (*cf.* §3.6.3). It also provides reasoning capabilities that handle uncertainty and fuzzy aspects of sensed context, as well as the lack of information (*cf.* §3.6.4 to §3.6.7). Finally, it supports the plug-in of additional reasoning techniques, including techniques for the formal validation of models [BZ12] and for context prediction [BZS09] (which were not presented here for the sake of brevity).

3.7.2 Ease of Use and Expressive Power

The Context Spaces Theory provides intuitive modeling tools that facilitate the interdisciplinary collaboration characterizing this research effort, by allowing researchers with different backgrounds to discuss using simple concepts.

Like any specification-based technique, the Context Spaces theory does not require *a priori* probability assessment about the evaluated phenomena [Pad06], which would present the drawbacks illustrated in §3.3.

Differently from other specification-based techniques, *e.g.*, ontology-based frameworks, the Context Spaces model is extremely simple to use and powerful at the time. With the Context Spaces model, it is possible to model the relevance of contextual dimensions for a situation and to take uncertainty and ignorance into account. The performance in terms of time required for reasoning is acceptable even when reasoning on complex Situation Spaces, involving numerous Context Attributes and articulated acceptable regions [BZ11a].

3.7.3 Management of Uncertainty

Differently from Bayesian approaches, the Context Spaces theory uses semantics of uncertainty and degree of occurrence for situations [BZ11a]. The Bayesian approach assumes that situations either occur or not and estimates the probability of occurrence. The Context Spaces theory, instead, uses semantics of uncertainty (fuzzy logic) and degree of occurrence. Fuzziness is a sort of vagueness and uncertainty, which is not represented using probability: probability is treated as a measure of the *undecidability* in the outcome of clearly defined and randomly occurring events, while fuzzy membership is usually concerned with the ambiguity or undecidability inherent in the *description* of the event itself [ANH07].

As we previously discussed, the context of the user cannot be modeled and recognized, so a theory that allows to represent the inherent fuzziness of situations is the ideal tool for this doctoral research. This will allow the resulting system to identify the cases characterized by high uncertainty and undecidability. Such capability allows avoiding the provision of unsuitable functionalities, as in those cases it is better to mediate the decision through interaction with users or to simply do nothing.

3.8 Conclusions

This chapter presented the challenges and solutions associated with the objective of being aware of context. Such capability constitutes the foundation of any Smart Home, as it allows to provide users with information and services that are suitable with respect to their activity.

For this, we illustrated the evolution of the context-awareness research domain. We showed that techniques evolved from a data-centered approach towards a situation-centered approach, allowing to simplify the development and maintenance of applications.

We presented existing techniques that allow to model contextual information and to reason about it to infer the occurrence of interesting situations. We evaluated such techniques with respect to their suitability to the requirements of this doctoral research. Namely, the ease of use, the expressive power and the offered tools to handle uncertain context were used as evaluation criteria.

Finally, we identified a technique that fulfills the requirements of this doctoral research, the Context Spaces Theory. Such technique offers a general-purpose framework for easily modeling complex real-world situations using intuitive concepts. It also includes reasoning tools that handle the uncertainty of sensed context and the inherent fuzziness of situation models.

The next chapter illustrates how to apply what we learned to the realization of sustainable domestic situations. More specifically, we design a Smart Home following the principles that emerged from our analysis of the context-awareness research domain and from our interdisciplinary design. Namely, applications adapt their behavior based on the occurrence of situations, which is detected by modeling and reasoning about contextual information using the Context Spaces Theory.

Chapter 4

Designing a Context-Aware System

The previous chapter provided some fundamental notions concerning the context awareness. We described the main challenges to solve and the existing approaches to tackle them. Finally, we introduced the Context Spaces Theory as a context-aware framework that offers ease of use, expressive power and ability to deal with uncertain context.

The goal is now to leverage the Context Spaces Theory to realize a context-aware system. This will have to enable the realization of sustainable domestic situations, the ultimate goal of this doctoral research.

This chapter aims at describing a complete context-aware system. For this reason, we have to broaden the scope of the presentation, as it was so far limited to the context modeling and reasoning techniques. To this end, we first illustrate the design principles and the basic technological choices for the system, in Section 4.1. Then, Section 4.2 illustrates the architecture of the system.

Some technical choices were required to allow the integration of the Context Spaces Theory in the overall system. Section 4.3 describes such choices. Furthermore, Sections 4.4 to 4.6 illustrate the improvements of the theory that were necessary to enable the fulfillment of all the goals of this doctoral research. Finally, Section 4.7 evaluates the compliance of the resulting system with the design principles presented in Section 4.1.

4.1 Design principles

This section illustrates the principles that were followed to design the context-aware system, constituting the computing architecture of the Smart Home. Such principles emerge from the human factor considerations, illustrated in Chapter 2, and from the analysis of the Smart Home and context-awareness literature, presented in Chapters 1 and 3.

4.1.1 Acceptability-driven Design

As discussed in Chapter 1, most existing Smart Homes were designed with a technology-driven approach. That is, the designers explored which services, functionalities, actions and controls could be performed exploiting available technologies.

One of the drawbacks of such an approach is that the acceptability of the involved technologies by final users is often not taken into account. In particular, solutions relying on heavy instrumentation (*e.g.*, wearable sensors, microphones, video cameras [CN09]) may be difficult to deploy and get accepted in real-world households, because of convenience and privacy concerns.

Many people have concerns on carrying equipment or feeling observed or recorded while living their private life [KCC+09]. This could seriously impact the acceptability of the Smart Home system or reduce its diffusion in real households.

Comfort/Privacy trade-off More specifically, there exists a trade-off between the benefits and drawbacks of a technological system that determines its acceptability for final users. In our specific case, the trade-off is between the suitability and usefulness of provided functionalities and the level of disruption of privacy that the system causes.

For an "average" family, the trade-off is often difficult to balance. People are often not keen on compromising their privacy, unless a high monetary or comfort benefit is obtained.

Minimum impact principle For this reason, we decided to take a very conservative approach, choosing technologies that are as unobtrusive as possible, in order to explore the frontiers of what can be done in a Smart Home with a very limited instrumentation.

We designed our system with an *acceptability-driven* approach. That is, we selected technologies that respond to the constraints of a real-world deployment of the future Smart Home system, namely, convenience and privacy concerns. Following the same considerations, the adopted technologies and techniques had to guarantee a fast and easy configuration, ultimately allowing a *plug-and-play* deployment.

Our positioning is on guaranteeing the minimum impact on the activity and house environment, while reaching the system goals: reducing energy consumption while maintaining or increasing inhabitant comfort. For this reason, we aim at designing functionalities that can be provided by maintaining a limited instrumentation and by minimally impacting or modifying the activity.

Augmenting appliances A kind of instrumentation that is already available in common households is represented by household appliances. These can be exploited to obtain contextual information, used to recognize ongoing situations. Since one of our goals is the efficient use of energy, the recognition of such situations will be used to decide how to manage appliances, so as to adapt their consumption to the real use that is made by people. For instance, when an appliance is left unattended, the system can provide the user with an interface to turn off the appliance remotely.

4.1.2 Successive Abstraction of Contextual Information

As we showed in Chapter 3, the recognition of complex human situation and activities should rely on successive abstractions of contextual information, from raw sensor data to successively higher levels, until the contextual information is sufficiently abstracted from low-level details [OHG02, CCDG05].

Raw sensor data should be directly used only to determine low-level contextual abstractions, *e.g.*, the presence or the posture of a person. Starting from this kind of contextual abstraction, we can determine higher level contextual information [YDM11]. For instance, starting from information about the presence of a person and combining that with contextual information provided by augmented appliances (*e.g.*, a smart television) the contextual information about the possible situation (*e.g.*, watching television), can be inferred.

4.1.3 Uncertainty and Ignorance Management

If contextual information has to be abstracted in successive steps, sources are not always reliable. In particular, uncertainty is intrinsic to the physical sensors that are used in the capture [DP01]. Thus, the uncertainty of lower abstraction layers will negatively impact the inference and decisions of the upper layers.

On the other hand, the acceptability and usefulness of the Smart Home can be achieved only if the system has an intelligible behavior. In particular, automated actions should be realized by the system only where there is a very high certainty that the situation asks for them. Otherwise they would be perceived as annoying and frustrating.

This means that the system must be aware of its ignorance, *i.e.*, recognize when the level of uncertainty is too high to provide adapted functionalities. When this happens, actions should be delegated to users.

An important issue arises: the management of uncertainty and ignorance. Information about uncertainty and ignorance has to be propagated, cumulated and considered at every abstraction step. Whenever the level of uncertainty becomes excessively high, the system should evaluate the trade-off between the potential benefit of providing the right functionality and the risk associated with an unsuitable functionality.

The next section presents an overview of the context-aware system that was designed and realized. We followed the principles that were just illustrated. Namely, the necessary equipment was chosen so as to allow privacy preservation and high acceptability; contextual information is successively abstracted and uncertainty, imprecision and ignorance are assessed and taken into account.

4.2 System architecture

We designed the system architecture drawing inspiration from the work of Coutaz *et al.*, who suggested a four-layer model to build context-aware applications [CCDG05], as shown in Figure 4.1.

The first layer, *Sensing*, is in charge of sensing the environment. The second layer, called *Perception*, realizes the abstraction from raw sensor data. These are processed to obtain more abstract information about the context (*e.g.*, the presence in a room). *Situation and Context Identification*, the third layer, identifies the occurring situations and the activities of inhabitants. For instance, the fact that a given moment a person is ironing can be modeled combining the information that a person is present in a room

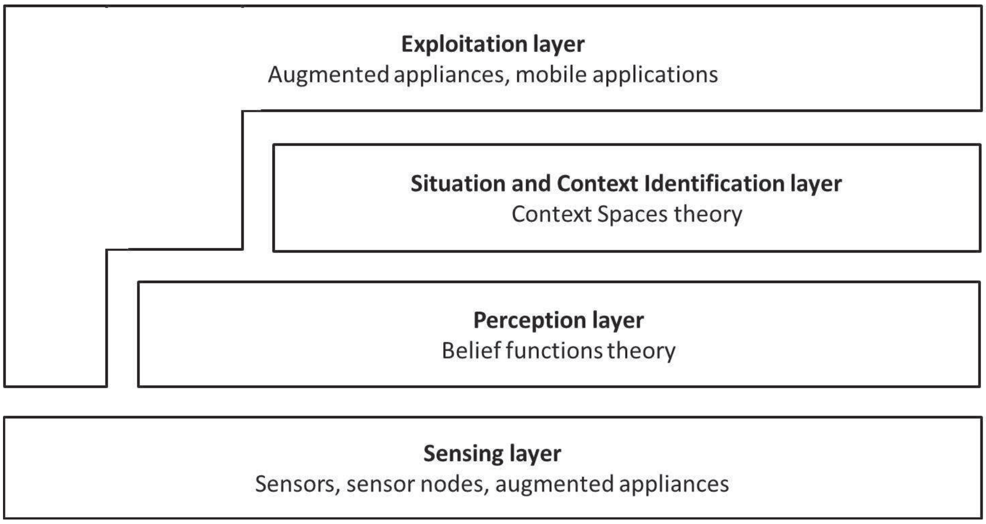


Figure 4.1: Four-layer model for context-aware applications (adapted from [CCDG05])

with the fact that the iron is on and that it is being moved. The top layer, called *Exploitation*, provides contextual information to applications.

An overview of the context-aware architecture is depicted in Figure 4.2. The rest of this section presents each layer of the architecture, describing the involved techniques and tools.

4.2.1 Sensing layer

The first layer of the architecture is made of *augmented* appliances and physical sensors.

Augmented appliances are household appliances equipped with computational capabilities and communication interfaces, which can publish meaningful information to other devices in their proximity and modify their behavior according to context [DZWB10]. For example, an augmented heater can provide information about its state (*e.g.*, goal temperature, consumption, heating mode), thus acting as a provider of contextual information, and turn itself off when nobody is present in the room for a certain time.

Given the recent trends in consumer electronics and the ongoing standardization efforts for the *Internet of Things* technologies, assuming the medium-term availability of such kind of augmented appliances in households seems realistic [Gra12]. This allows fulfilling the identified requirements in terms of privacy and acceptability, while still providing meaningful and reliable contextual information and functionalities.

As a supplement of contextual information obtained from appliances, physical sensors are dispatched in the environment, in order to capture additional kinds of information that are not otherwise obtainable. Some raw data are immediately exploitable, like temperature or light level. Others require processing in order to obtain more abstract contextual information, such as inhabitants' presence. The Perception layer, presented below, realizes such task.

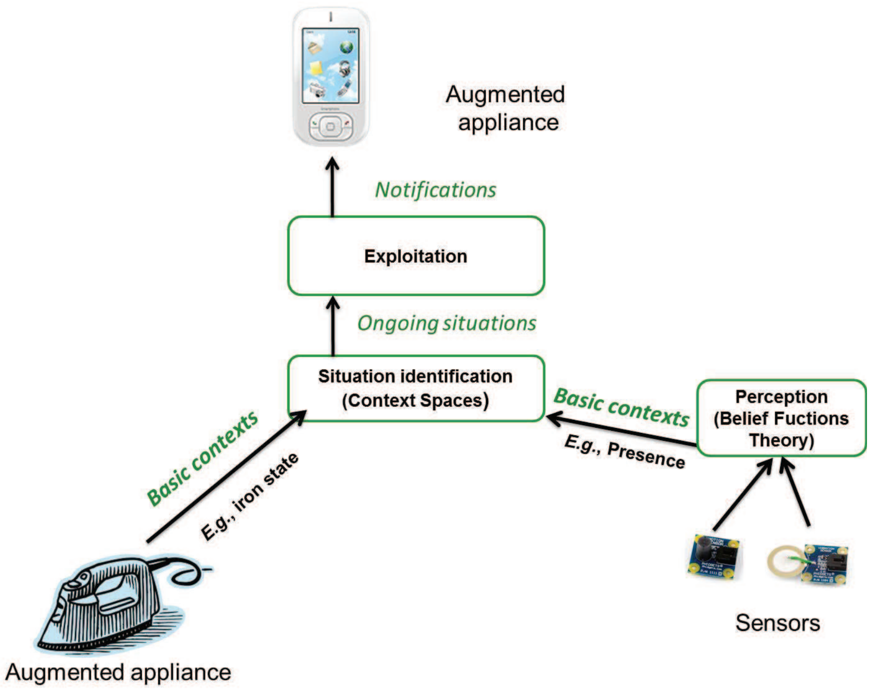


Figure 4.2: The architecture of the context-aware system

4.2.2 Perception layer

In the Perception layer of the proposed model, raw sensor data are processed to obtain more abstract information about context, easily understandable by humans and by the upper layer (*e.g.*, the presence or number of people in a room, the posture of someone). Such process is called *sensor data fusion*.

Sensor data fusion is a difficult problem. This is due to the several reasons, all related to imperfections characterizing sensor data: randomness, inconsistency, incompleteness, ambiguity, uncertainty, bias, redundancy, *etc.* [DP01]

For this reason, the context-aware system includes a dedicated framework to perform sensor data fusion, the *belief functions theory* [Sha76]. The choice of such theory and the adaptation required to integrate it in the Perception layer were realized as part of Bastien Pietropaoli's doctoral research [PDW11, PDW12, DPW12].

4.2.3 Situation and Context Identification layer

Having abstracted from the raw sensor data, the system has to reason about context, in order to infer higher-level context information, needed to make decisions concerning the functionalities to offer to inhabitants.

As we previously illustrated, our context-aware system has to provide targeted functionalities that enable the transition from existing to sustainable situations. To this end, the goal of the system is first of all to recognize the contextual configurations that suggest the occurrence of existing situations to transform.

To realize this task, we adopted the intuitive context and situation modeling and reasoning mechanisms offered by the Context Spaces Theory (CST), described in the previous chapter.

We use the CST to implement the Situation and Context Identification layer of our architecture. We model interesting situations as combinations of basic contextual information provided by both the sensor data fusion (executed by the Perception layer) and augmented appliances.

4.2.4 Exploitation layer

As explained by Coutaz *et al.* [CCDG05], the Exploitation layer acts as an adapter, allowing applications to address to the infrastructure their requests for context services at a high level of abstraction. In our architecture, this layer provides information about context to the applications running on augmented appliances, which adapt their behavior in a semi-automatic way and allow non-interruptive interaction and takeover by inhabitants.

More specifically, leveraging the Exploitation layer, applications can be notified about the occurrence of situations. The context-aware system described in this chapter is able to reason on the context in order to determine the occurrence of situations. Applications can subscribe to the situations they are interested in and be notified as soon as those situations occur. This allows applications to adapt their behavior depending on context and situations.

To realize the Exploitation layer, we designed and developed some software components that allow applications to subscribe to the situations they are interested in. The reader can refer to Chapter 5 (and Section 5.5 in particular) for a description of the design and implementation choices.

The current chapter continues by going into details of the adoption and improvement of the Context Spaces Theory. In particular, the next section shows the novelty of our approach in using the theory as a component of a complete context-aware system.

Then, Sections 4.4 to 4.6 illustrate the modifications and improvements of the theory that were realized in the context of this doctoral research. These were required in order to allow the flow of information about uncertainty and ignorance between the layers of the architecture (*cf.* Sections 4.4 and 4.5). Furthermore, we introduced the capability of modeling and recognizing the temporal dimension of real-world situations (*cf.* Section 4.6).

4.3 Adopting the Context Spaces Theory

We use the Context Spaces Theory (CST) to implement the Situation and Context Identification layer of our architecture. We model interesting situations as combinations of basic contextual information provided by both a sensor-data-fusion technique (executed by the Perception layer) and by augmented appliances. Adapted functionalities are provided when the interesting situations are triggered. The recognition of ongoing situations is made possible by the CST through reasoning about available context information.

This section presents a fundamental novelty that we introduced with respect to previous applications of the Context Spaces Theory: a different level of abstraction, due to the position of the CST in our context-aware architecture. The section also highlights the potential that is generated by this novelty: the flow of a measure of uncertainty and ignorance.

4.3.1 A different level of abstraction

Unlike previous approaches [BZ12, BZ11a, BZS09, BZ11b, DHKZG08, PZL06], we do not use the Context Spaces Theory to directly deal with sensor data. As presented in §3.6.2, in the original theory, Context Attributes and their values are directly provided by physical or virtual sensors. In our architecture, raw sensor data are preprocessed by the Perception layer using a dedicated sensor data fusion framework. This allows to abstract from raw sensor data and to perform reasoning about situations leveraging higher-level contextual information.

More specifically, the Context Attributes' values at time t (denoted by a_i^t) are either provided by the augmented appliances or output by the sensor data fusion framework. An example of Context Attribute provided by an appliance, *e.g.*, the radio, indicates its state, which can either be $\{ON\}$ or $\{OFF\}$. An example of Context Attribute provided as output of the sensor data fusion is *Posture*, with the following possible values: $\{Seated\}$, $\{Standing\}$, $\{LyingDown\}$.

Such novelty of our approach with respect to previous research efforts is presented in Figure 4.3.

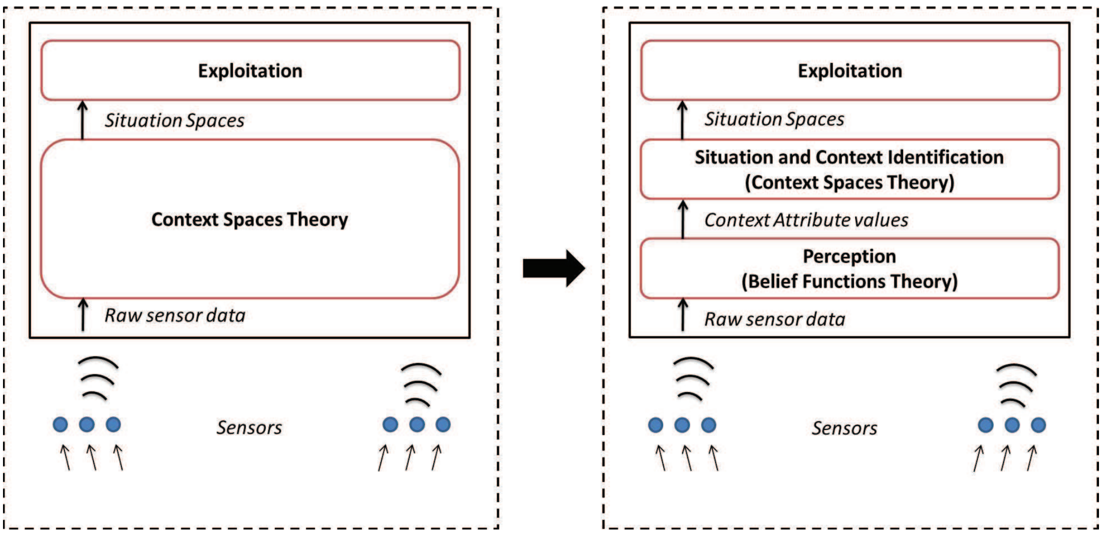


Figure 4.3: A novel use of the Context Spaces Theory

4.3.2 Exchanging information with the Perception layer

The fact that Context Attributes are provided to the Situation and Context Identification layer by the Perception layer generates new possibilities. The two layers, indeed, can exchange information that enriches the reasoning and improves the recognition of situations. To understand such potential, we need to introduce more details about the Perception layer and about the belief functions theory.

The Perception layer aims at combining different sensor readings so as to calculate the runtime value of interesting Context Attributes. This process is called sensor data fusion and, in our architecture, it is realized using the Belief Functions Theory (BFT).

To produce a value for a Context Attribute starting from sensor readings, the BFT follows several steps, illustrated below [DPW12].

Define the possible worlds

In the BFT, the first thing that should be defined is a set of possible worlds $\Omega = \{\omega_1, \omega_2, \omega_3, \cdots, \omega_n\}$ called the *frame of discernment*. These worlds have to be exclusive and if possible exhaustive. For example, the set of the possible postures of someone can be defined as $\Omega = \{Seated, Standing, LyingDown\}$. These will be the possible values of the Context Attribute *Posture*.

From this set of possible worlds, the powerset is created using the disjunctive operation. Thus, we obtain:

$$2^\Omega = \{\{\omega_1\}, \{\omega_2\}, \cdots, \{\omega_1 \cup \omega_2\}, \cdots, \{\omega_1 \cup \omega_2 \cup \omega_3\}, \cdots\},$$

the set of all possible subsets. For example, the posture of someone could be a set such as $\{Seated \bigcup LyingDown\}$. This expresses imprecise knowledge, *i.e.*, a case where the system cannot decide which one is the true state of the world, so it only indicates a set of states, specifying that the true state is one of those.

Build the mass functions

Once the frame of discernment is created and the corresponding powerset generated, a *mass function* (also called *basic belief assignment* or *body of evidence*), representing the degree of belief associated to each subset of Ω, is defined such that:

$$m : 2^\Omega \to [0, 1]$$
$$\sum_{A \subseteq \Omega} m(A) = 1$$

With this definition, it is easy to see that mass functions offer a double way to express uncertainty: with degrees of belief and with sets of possible worlds.

Every subset A with $m(A) > 0$ is called a *focal set* and may be considered as a part of belief. The mass accorded to the complete set of possible worlds ($m(\Omega)$) is called *total ignorance* and represents the degree of belief accorded to the fact that the system has absolutely no clue on what is going on. This total ignorance is really important in our system which should be able to not disturb inhabitants with non-adapted services.

Build and fuse the evidence

The belief functions theory works by accumulation of evidence. The more evidence gathered the more certain and precise the final result should be. In our case, evidence is provided by sensors. For instance, the detection of motion by a sensor indicates that there might be someone in the room. If, in addition to motion, other sensors detect vibrations and sounds, the evidence of the presence of somebody increases.

Once the evidence is gathered from sensors, all the mass functions are fused in order to get the global belief of the system. To do this, it is possible to use one of the several existing *rules of combination* [SK94, Yag87, DP88, Mur00, CWkYZf05, MO08], which we will not present here for the sake of brevity.

Decision making

Once the evidence fused, the resulting mass function is hard to interpret by itself. Remember that the output of the Perception layer is used by the Context Spaces Theory, which requires as input values for Context Attributes. Thus, a decision has to be made by the Perception layer, so as to output something exploitable by the upper layers.

The decision-making can be realized using the maximization of different criteria. In the literature, there exist three common criteria. They are called *belief* (or *credibility*), *plausibility* and *pignistic transformation* (or *bet on the probability*) [Sme05, DPW12].

The credibility may be interpreted as the degree of certainty about something that is occurring or the probability of provability associated to a subset of Ω [DPW12]. It is seen as a pessimistic criterion as it corresponds to the mass given directly to the subset of possible worlds and the subsets of this subset. The plausibility can be interpreted as the support accorded to the fact that something is possible. It is seen as an optimistic criterion as it corresponds to the maximum mass that could potentially be accorded to the subset of possible worlds. The bet on the probability is a neutral criterion as it equally distributes the mass between the different possible worlds.

The decision-making is realized by choosing the possible world that maximizes one of the presented criteria. In our context-aware system, we mainly used the bet on the probability, as it is a neutral criterion. For example, the decision about the posture of somebody is taken by selecting the posture that maximizes the bet on the probability criterion, given the current sensor readings.

To summarize, the Perception layer uses evidence provided by sensors to determine the most likely value for a Context Attribute. The output of this layer is thus the runtime value of a Context Attribute.

4.3.3 A new potential: the flow of uncertainty

We illustrated how the Perception layer produces Context Attribute values starting from evidence provided by sensor readings. Once the Context Attribute value has been calculated, it can be provided to the Context Spaces Theory.

When presenting the requirements of our context-aware system, we highlighted that the management of uncertainty at the system level is an important requirement. In the current description of the architecture and of the exchange of data between layers, the management of uncertainty is not considered.

However, both the Belief Functions Theory and the Context Spaces Theory allow to keep track and take into account uncertainty when performing their tasks. Coupling the tools of the two theories for the management of uncertainty can generate new possibilities for a cross-layer (and, thus, system-level) management of uncertainty.

As we previously illustrated, the decision-making at the Perception layer is realized by maximizing a function that estimates the credibility, the plausibility or the bet on the probability that the Context Attribute value is a particular one (*cf.* §4.3.2).

In other words, the function calculates an assessment of the *confidence* of the Perception layer in the fact that the returned value is the correct one. Providing such confidence value to the Situation and Context Identification layer means realizing a flow of confidence between layers, as depicted in Figure 4.4.

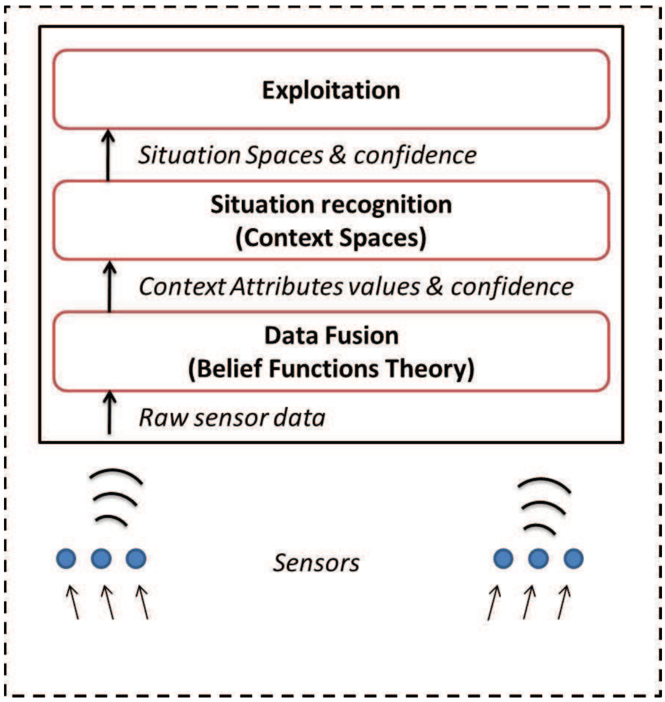


Figure 4.4: Cross-layer confidence flow

The new possibilities opened by the communication between layers ask for modifying or making particular uses of the original tools offered by the Context Spaces theory and the belief functions theory. The next sections show the improvements that were realized to the Context Spaces theory.

4.4 Reconsidering the management of uncertainty

This section shows how we realized a cross-layer management of uncertainty, based on the communication of confidence values associated to Context Attributes. We first remind the basic tools offered by the Context Spaces Theory and show their limitations. Then, we present an improved heuristic that solves the highlighted limitations. Finally, we briefly illustrate the benefits of the cross-layer uncertainty propagation.

4.4.1 Handling Uncertainty with the CST

We now recapitulate the tools provided by the CST for handling uncertainty; then, we show the limitations of existing heuristics. Finally, we illustrate an improved heuristic that allows to propagate confidence information between our layers.

Reasoning about situations As explained in §3.6.4, in order to infer the occurrence of a situation at a given moment, the Context Spaces reasoning uses a two-step process. First, the *degree of confidence* in the occurrence of a situation is calculated through a *matching function* μ. Then, the occurrence of the situation is calculated by a Boolean *inference function* $\gamma = (\mu \geq \varepsilon)$, where $\varepsilon \in [0, 1]$ is a confidence *threshold*. In other words, a situation is ongoing if its confidence, calculated using the heuristics, exceeds a fixed threshold.

Sensor reliability and inaccuracy In the calculation of the confidence in the situation, the matching function μ is implemented so as to take into account the sensor reliability and inaccuracy, using Equation 3.6.3, which we write down for memory:

$$\mu\left(C^t, S, \sum\right) = \sum_{i=1}^{n} w_i Pr\left(\hat{a}_i^t \in A_i\right), \tag{3.6.3}$$

where $Pr\left(\hat{a}_i^t \in A_i\right)$ represents the probability of a Context Attribute's correct value (denoted by $\hat{a}_i^t$) being contained within the relative acceptable region A_i (*cf.* §3.6.5).

This heuristic provides an approach to compute the contribution level of a Context Attribute value at runtime, based on the probability of sensor reliability or its accuracy. The benefit of this heuristic is that it gives higher weight to the Context Attributes provided by more reliable sensors. It also allows taking into account sensor inaccuracy, provided that an estimation of reading error distribution is available.

The limitation of this heuristic is that it deals with the probability that a Context Attribute value is *contained* in an acceptable region, regardless of where exactly inside the region the value may fall.

Fuzzy Situation Inference As explained in §3.6.6, the Fuzzy Situation Inference (FSI) technique solves this drawback by reflecting delta changes of Context Attribute values in the situation inference results. To this aim, the heuristic calculates the confidence in a Situation Space S at time t as:

$$\mu(C^t, S, \sum, E) = \sum_{i=1}^{n} w_i \varphi_i\left(f(a_i^t, e_i^t)\right), \tag{3.6.5}$$

where φ_i is the membership degree of the context attribute value a_i^t after that it has been corrected using the function $f(a_i^t, e_i^t)$. This function calculates the correct value of the context attribute based on the error value e_i^t. If e_i^t is a reliability rate, a_i^t is multiplied by it. Conversely, if e_i^t is an inaccuracy measure, it is added to a_i^t [DHKZG08].

4.4.2 Drawbacks of existing management of uncertainty

Neither the original CST heuristics nor FSI offer adequate tools to handle the uncertainty of the Context Attribute values provided by the Perception layer of our architecture. The reasons are detailed below.

Drawbacks of estimating the accuracy of sensors The estimation of sensor reading error distribution, used by the CST to handle uncertainty, is not useful because the Perception layer preprocesses sensor readings. Readings coming from several sensors can be fused together and the statistical distribution of the error of individual sensors becomes useless.

Estimating the accuracy of the output of the Perception layer is also not possible. The sensor data fusion performed by the Perception layer, indeed, provides as output one or more discrete values (see §4.2.2). The notion of inaccuracy is thus not relevant, as the value itself is not selected from an infinite, continuous set.

Drawbacks of estimating the reliability of sensors Even the estimation of the reliability of sensors offered by the CST is not suitable for taking into account the uncertainty of Context Attribute values.

As already mentioned, the original formulation of the CST uses the reliability of sensors to calculate the probability that the Context Attribute value falls within the acceptable region. This neutralizes the benefits of the contribution function, as it eliminates the possibility to take into account the exact point of the region in which the value falls.

Even FSI does not offer proper means to take into account the reliability of Context Attribute values. In FSI, indeed, the function $f(a_i^t, e_i^t)$ multiplies the value of the Context Attribute by its reliability *before* applying the membership function $\varphi_i\left(f(a_i^t, e_i^t)\right)$ (*cf.* Equation 3.6.5). As the Perception layer only produces non-numerical values as output, multiplying the Context Attribute value by a reliability rate does not make sense.

In order to improve the management of uncertainty in our architecture, we need to introduce an additional heuristic, illustrated below.

4.4.3 Improved Management of Uncertainty

We introduce a new heuristic to take into account the uncertainty of Context Attribute values. The resulting matching function, used to calculate the confidence in the occurrence of a situation S at time t, is formalized as follows:

$$\mu(C^t, S, \sum, \Delta) = \sum_{i=1}^{n} w_i \kappa_i(a_i^t)\delta_i^t, \tag{4.4.1}$$

where $\kappa_i(a_i^t)$ and δ_i^t are the contribution function and the *confidence* associated to the Context Attribute value a_i^t, respectively. κ_i is defined at design-time as part of the Situation Space model (*cf.* §3.6.3), while δ_i^t is provided at runtime by the Perception layer (*cf.* §4.3.3).

4.4.4 Benefits of the improved management of uncertainty

When applied to our architecture, the newly introduced heuristic combines the benefits of the original CST formulation (*cf.* Equation 3.6.1) with the additional capability of taking into account the runtime reliability of Context Attribute values.

Existing techniques for handling uncertainty with the CST correct the Context Attribute value *before* applying the contribution or membership functions. Instead, we correct the *output* of the contribution function with the confidence value provided by the Perception layer. This is more sensible in our specific case, for two reasons.

First, the Context Attribute values are not real values but elements of a set, so 'correcting' their values is not an applicable concept.

Second, we want to have a different contribution of the Context Attribute depending on where exactly the value falls inside the acceptable region; so, using a probability of containment is not enough. We thus need to combine both the contribution function $\kappa_i(a_i^t)$ (*cf.* §3.6.3) and the correction depending on the confidence value δ_i^t, which can be provided by the Perception layer at runtime.

The degree of confidence on the occurrence of the situation, as obtained using the previous expression, has the property of being reduced when the confidence on the individual Context Attribute values is reduced. In this way, a very low confidence on the value of a Context Attribute can prevent the confidence threshold of the situation from being reached and thus the situation from being triggered. Uncertainty is thus taken into account and used to avoid providing unsuitable functionalities.

4.5 Considering Imprecision and Ignorance

When presenting the design principles of our context-aware system (*cf.* §4.1), we highlighted the need to consider ignorance as a possible outcome of the context processing.

A system that is aware of its ignorance can perform better decisions, for two reasons. The first one is that this capability allows to detect when the available information is not sufficient to make a safe decision about the functionality to provide. This helps preventing the system from providing unsuitable functionalities.

The second reason is that, despite ignorance, the system might be able to make an overall clear-cut decision. For example, to understand that someone is sleeping, sometimes it is sufficient to know that the person is seated or lying down and not standing. So, tolerating ignorance can improve the recognition process.

In our system, the Perception layer can transfer a non-specific (*i.e.*, imprecise) result to upper layers. The goal is now to design a solution to transfer such imprecise knowledge to the Context Spaces theory, as depicted in Figure 4.5.

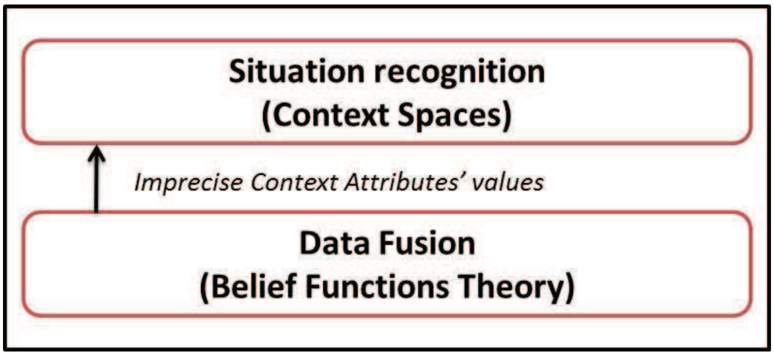


Figure 4.5: Cross-layer uncertainty, imprecision and ignorance propagation

In this section, we illustrate such a solution. For this, we first illustrate the mechanism that allows to provide imprecise values as output of the Perception layer. Then, we provide a solution to enable the Context Spaces theory to take into account imprecise knowledge and ignorance. The result will be a system that detects when safe decisions cannot be taken and, conversely, when imprecise knowledge and ignorance are not a problem.

4.5.1 Imprecise output of the Perception Layer

At the Perception layer, ignorance can be due to lack of information, *e.g.*, because of losses in the communication of sensor readings. It can also be due to imprecise knowledge, because the available information does not allow making a clear-cut decision. For example, the fusion of sensor data at the Perception layer might be unable to determine if a person is sitting or lying down, although knowing that the person is not standing.

As we previously illustrated, the decision-making at the Perception layer is realized by maximizing a function that estimates the likelihood that the Context Attribute value is a particular one (*cf.* §4.3.2).

As we said, it could be more interesting in the case of the posture of someone to get a high certainty on the subset $\{Seated \bigcup LyingDown\}$ than a lower certainty on one of the atomic subsets $\{Seated\}$ and $\{LyingDown\}$. We might thus relax the constraint saying that the returned Context Attribute value must have cardinality equal to one.

However, the criteria for the decision-making at the Perception layer increase with the cardinality of the subsets (*cf.* §4.3.2). Thus, the criteria's maxima are always obtained with the complete set of possible worlds (Ω). That is, being free to increase the cardinality would lead the Perception layer to always output an absolute certainty of being ignorant. A compromise between certainty and precision is necessary.

In order to obtain that compromise, the Perception layer implements a *filtering* algorithm [DPW12]. The algorithm uses a threshold to decide when the certainty is sufficient to consider the answer as true. That is, the cardinality of the Context Attribute is increased until the minimum confidence threshold is reached. It may happen, in some situations, that the system cannot be sufficiently sure of anything. In this case, it outputs Ω with a certainty equal to 1, meaning that it does not know what is happening. The filtering algorithm is illustrated in Figure 4.6.

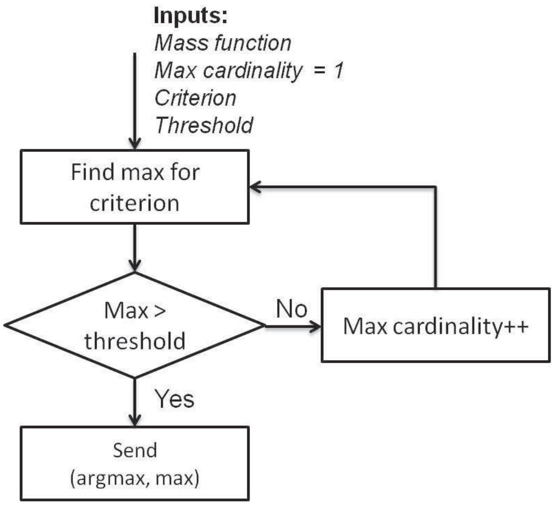


Figure 4.6: The *filtering* algorithm

To summarize, the *filtering* algorithm, developed at the level of the Perception layer, enables the required exchange of information between layers and also allows the flow of an estimation of uncertainty, imprecision and ignorance. This provides as output both a decision concerning the value of the Context Attribute and a measure of confidence on that decision. The value can also be imprecise, meaning that the filter can return a set of values instead that a single value. For instance, to inform that a person is most likely seated or standing (but not lying down), the Perception layer will provide the set of values $\{Seated \bigcup Standing\}$.

4.5.2 Existing tools to Handle Imprecision and Ignorance

As illustrated in §3.6.7, the Context Spaces Theory offers some tools to deal with ignorance; namely, the case of missing Context Attributes at runtime. When it comes to imprecise knowledge, the theory offers the Fuzzy Situation Inference (FSI) technique.

As explained in §3.6.6, using the FSI technique, a Situation Space is defined by multiple conditions joined with the *AND* operator, where each condition can itself be a disjunction of conditions (*i.e.*, using the *OR* operator) over fuzzy sets [DHKZG08]. For instance, the simple situation of 'comfortable room' can be defined as the combination of the following facts: the room temperature is *warm AND* the room light level is *bright AND* the sound level is *low*.

In the definition of the situation 'comfortable room', *warm* is a fuzzy variable. The actual value provided by a sensor is mapped to such fuzzy variable by a membership function. Intuitively, a warm temperature (*e.g.*, 20° Celsius) will have a high membership degree for the fuzzy variable *warm*, while a low temperature (*e.g.*, 10° Celsius), will have a very low membership degree for that same fuzzy variable.

So, a Context Attribute contributes to a Situation Space with a different contribution depending on the membership degree of the value to a fuzzy set.

4.5.3 Drawbacks of the Fuzzy Situation Inference (FSI) technique

Unfortunately, FSI presents some drawbacks when applied to our architecture. Namely, it does not allow communicating imprecision from the Perception layer to the Situation and Context Identification layer. This is due to the fact that the fuzziness that is addressed by FSI is at the level of sensor readings. Instead of requiring designers to model exact values of sensor readings when defining the acceptable regions for Context Attributes, FSI allows to only specify situations as combinations of fuzzy terms (*e.g.*, *warm* instead that $18 \leq temperature \leq 22$ for the temperature in the room).

In our architecture, the Context Attributes already express some real-world concept (*e.g.*, the occupation of a room). Defining an extra layer of fuzziness over the Context Attributes values does not help solving the issue of imprecision and uncertainty propagation. For this reason, FSI is not suitable for our purposes.

We now show how we can instead use the basic Context Spaces theory to reach our goal. To this end, we illustrate how the imprecise output of the Perception layer can be provided as input to the Context Spaces Theory, so that the existing reasoning tools offered by the theory can be reused.

4.5.4 A novel approach to imprecise reasoning with the CST

In order to propagate imprecise information and ignorance from the Perception layer to the Situation and Context Identification Layer, we define additional values for each Context Attribute, so that the new set of possible values for a Context Attribute is the *powerset* of the original set. A powerset is the collection of all the possible subsets of the original set.

For instance, the set of possible values of the Context Attribute *Posture* can be extended as follows:

$$a^t_{posture} \in \left\{ \begin{array}{c} \{Seated\} \\ \{Standing\} \\ \{LyingDown\} \\ \{Seated \bigcup Standing\} \\ \{Seated \bigcup LyingDown\} \\ \{Standing \bigcup LyingDown\} \\ \{Seated \bigcup Standing \bigcup LyingDown\} \end{array} \right\}$$

4.5.5 Benefits of the novel approach

Considering the powerset of the basic set of possible values of a Context Attribute, one can define a finer-grained contribution function. For instance, the values $\{Seated\}$, $\{LyingDown\}$ and $\{Seated \bigcup LyingDown\}$ can all strongly contribute to a situation *Sleeping*, while $\{Seated \bigcup Standing\}$, $\{Standing \bigcup LyingDown\}$ and $\{Standing\}$ can have decreasing contributions. Finally, $\{Seated \bigcup Standing \bigcup LyingDown\}$ can either be modeled as a zero-contribution value or be assimilated to a missing Context Attribute (causing the weights of the other Context Attributes involved in the situation to be recalculated — *cf.* §3.6.7).

The existing reasoning techniques can be applied without modifications, provided that the contribution function assigns a contribution to every possible combination of individual values of a Context Attribute, as in the example illustrated above.

Using the presented approach to the management of imprecision and ignorance, we allow a fine-grained modeling of situations, taking into account uncertainty, imprecision and ignorance. This improves the flexibility of the modeling tools offered by the Context Spaces theory, while at the same time reusing the existing reasoning techniques.

For the sake of the overall system architecture, this mechanism allows imprecision and total ignorance (which, in the example, are represented by a set of cardinality 2 and 3, respectively) to be communicated from the second layer to the third layer. When the Perception layer returns multiple possible values of a Context Attribute, the corresponding contribution is used in the CST reasoning process. This achieves the flow of imprecise values and ignorance between layers, as it was depicted in Figure 4.5.

4.6 Integrating a Temporal Dimension

We presented the novel approaches to the management of cross-layer uncertainty, imprecision and ignorance flow. We now introduce another interesting functionality, allowing to model and reason about the temporal dimension of situations.

As we saw in the first part of the dissertation, the gap between human context and recognition capabilities of any context-aware system asks for handling the ambiguity of the observed behavior of people. More specifically, we saw that even a very simple real-world scenario can hide great complexity and uncertainty over the action to perform. In particular, we presented the example of someone entering a room, demonstrating that it is impossible to guess if they will stay long time or quickly leave (*cf.* §2.5.1).

In order to design functionalities that take into account the possible actions of the person, which directly depend on their current engagements and concerns (which are not observable), we need to introduce the temporal dimension in the Context Spaces Theory. The temporal dimension, just like any other dimension in the Context Spaces model, allows accumulating evidence for the recognition of a situation. Coming back to the scenario, the situation representing a stable occupation of a room by a person can be recognized with more and more confidence as time passes.

This section shows that the CST does not provide suitable tools to model and reason about the temporal dimension of real-world situations. We then describe the solution that we designed to remedy to this shortcoming.

4.6.1 Lack of tools to handle the temporal dimension in the CST

The temporal dimension is considered as any other dimension in the original Context Spaces Theory. For this reason, no particular tools are available to manage it.

Some research efforts have investigated the potential of predicting future context and acting proactively in reply to the prediction [BZS09, BZ10]. However, the tools developed by such efforts do not allow to model time constraints, nor do they allow specifying temporal aspects of situations.

Without specialized tools, it is not possible to model most interesting real-world concepts that involve time. Coming back to the example, we need to know for how long a room has been occupied. As we previously explained, our architecture provides as input to the Context Spaces theory basic contextual abstractions (the Context Attributes) that are obtained by fusing sensor readings at a given moment. For instance, the Context Attribute *Presence* will indicate the occupation of a room at a given moment. No information about the duration of such occupation is provided.

4.6.2 History Keepers: novel tools to handle the temporal dimension

To solve this problem, we introduced a new set of tools that can handle the temporal dimension in the Context Spaces Theory. These tools are called *History Keepers*.

The role of such tools is to dynamically generate new Context Attributes that keep track of some particular aspect of the history of a monitored Context Attribute.

Examples of the historical properties of a Context Attribute are its average value over time, the duration of the current value (*i.e.*, for how long the current value has been observed) or the last time a particular value was observed.

4.6.3 Advantages of the History Keepers

The advantage of the proposed tools is that the dynamically generated Context Attributes can be processed just like any "real" Context Attribute (*i.e.*, provided by the Perception layer). In this way, we can include in a situation model a dimension representing the temporal development of an interesting Context Attribute.

Coming back to the example mentioned above, we can now model the situation of stable presence by including an acceptable region for the duration of the occupation of the room. More specifically, the duration of the value {*yes*} for the Context Attribute *Presence* might be set to "more than 1 minute", meaning that the stable occupation situation is ongoing only after one minute of continuous occupation of the room. Such situation model is made possible by a *duration* History Keeper.

More details about the History Keepers will illustrated in the chapters dedicated to the implementation and evaluation of the system (*cf.* §5.4 and §6.3).

4.7 Benefits of the Proposed System

We illustrated of our system architecture and our contributions to the Context Spaces Theory. In this section, we demonstrate the benefits of the resulting context-aware system, with respect to the design principles illustrated at the beginning of this chapter and, more generally, to the goals of this doctoral research.

4.7.1 Acceptability of the System

The adopted instrumentation facilitates user acceptance and privacy preservation. We only use simple environmental sensors and exploit augmented household appliances.

We do not impose requirements or limitations on user behavior. For instance, no wearable sensors are required for the system to work.

Finally, we use specification-based techniques. These allow designing the deployment of the system and the applications “off-line”. There is no need for long learning phases once the system is deployed, which are likely to bother users.

4.7.2 Cross-layer Management of Uncertainty

The context-aware system is able to propagate uncertainty, imprecision and ignorance from layer to layer. This allows assessing the confidence in the final inference results.

Exploiting this estimation of the confidence, applications can make decisions concerning their behavior in a more informed way. Namely, by recognizing when the confidence is not high enough to guarantee trustworthy inference.

This can be modulated through the notion of risk. The risk assesses the gravity of the consequences of providing an unsuitable functionality. In case of high risk, one can state that the functionality is provided only when the confidence is very high. It is also possible to delegate to users, in case an action is required.

4.7.3 Simple Design

Designing context-aware applications and situations that trigger the provision of functionalities is simple and immediate. This is due to the intuitive context modeling and reasoning tools provided by the Context Spaces Theory, which are based on geometrical metaphors and simple heuristics.

The interdisciplinary approach that characterizes our research project can benefit from the use of these simple and intuitive tools.

4.8 Conclusion

This chapter illustrated the proposed context-aware system. We focused on the choices concerning the organization of the system architecture and on the description of the tools for modeling and reasoning about context and situations.

We first described the adopted design principles, including considerations about acceptability, successive abstraction of contextual information and management of uncertainty.

We then defined a layered architecture and presented the algorithms and tools that compose it. The system relies on lightweight instrumentation, only including augmented appliances and environmental sensors. The adoption of the Context Spaces Theory facilitates the interdisciplinary collaboration aiming at designing user-driven functionalities.

Finally, we presented the contribution of this doctoral research to the Context Spaces Theory. For this, we introduced novel tools and solutions that allow to take into account and propagate information about uncertainty and imprecision of context, as well as to model and recognize the temporal dimension of real-world situations.

The next part of the manuscript illustrates the choices that we made when implementing the presented context-aware system. We will provide an overview of the deployment choices and present the adopted implementation of the Context Spaces Theory, as well as the modifications that were made to such an implementation. Finally, we will present an example of context-aware application that we realized and evaluate the proposed tools.

Part III

Validating the Approach

We analyzed our technical requirements and presented a context-aware system that enables the provision of adapted functionalities in the domestic setting.

In this part of the dissertation, the last one, we present the implementation and deployment choices of our Smart Home. This is discussed in Chapter 5.

We also validate such choices and the overall contribution of this doctoral research. This is done in Chapter 6, which illustrates a context-aware application that realizes a sustainable situation. The motivation and specifications of such application originate in our interdisciplinary design, presented in the first part of this dissertation.

Chapter 6 also evaluates the context-aware system, by presenting some issues that arise when deploying the application on a prototype. These are mainly caused by the uncertainty of the sensing technology. Finally, we show that the tools designed and developed during this doctoral research provide effective solutions.

Chapter 5

Realizing the Proposed System

This chapter illustrates the main choices concerning the implementation and deployment of the system described in the previous chapter.

To this end, we first provide an overview of the implementation of the whole architecture (in Section 5.1.1). Then, we focus on the *Situation and Context Identification* and *Exploitation* layers of the architecture, as this doctoral research mainly concentrated on those aspects.

For this, Section 5.2 describes ECSTRA, the implementation of the Context Spaces Theory that we adopted. Then, Sections 5.3 to 5.5 illustrate the modifications and extensions of ECSTRA that were performed in the context of this doctoral research. Finally, the communication mechanisms between layers are presented in Section 5.6.

5.1 Overview of the Implementation and Deployment

We now present an overview of the implementation and deployment choices of our context-aware system.

To deploy the system, we have built a Smart Home demonstrator made of a kitchen and a living room. We will now use the occupation of such rooms as an example of contextual dimension, showing how it is obtained and exploited, layer by layer, across the context-aware architecture.

5.1.1 Sensing layer

As already illustrated, the first layer of the system is made of *augmented* appliances and physical sensors.

In our deployment of the system, augmented appliances include smartphones, heaters, hotplates, touch-screens and radios. All devices publish information about their state to other devices in their proximity. They are also augmented with context-aware capabilities, as they run an implementation of the Context Spaces Theory (*cf.* §5.1.3).

As a supplement of contextual information obtained from appliances, physical sensors are dispatched in the environment, in order to capture additional kinds of information that are not otherwise obtainable. Sensors return as output measurements of physical phenomena, expressed as numerical values.

Several sensors are grouped in wireless sensor nodes. Sensor nodes are components made of a microcontroller, a radio transceiver and an acquisition module for obtaining sensor readings. In our deployment, we adopted Zolertia sensor nodes[1]. As for the sensors, we used Phidget sensors[2].

Sensor nodes periodically report sensor readings to a sink using IEEE 802.15.4/6LowPAN communications. The sink is implemented by a *Plug Computer*. Plug Computers[3] are small computers that easily go unobserved in a normally furnished house, as they can be hung on the electrical socket. They are equipped with Ethernet, Wi-Fi and other input/output interfaces. In our deployment, we augmented Plug Computers with an IEEE 802.15.4/6LowPAN radio interface, so that they can communicate with the sensor nodes.

Figure 5.1 depicts the Sensing layer deployment. It shows a *GuruPlug* (a particular model of Plug Computer), augmented with a Raven USB stick[4] (an IEEE 802.15.4/6LowPAN radio interface). The left-hand side of the figure shows a Phidget *USB interface kit*. This enables the direct connection of sensors to a Plug Computer. The right-hand side shows two sensor nodes, wirelessly connected to the Plug Computer. Solid lines indicate wired connections, while wireless connections are depicted as broken lines. Augmented appliances are not represented in the figure, as they send the information about their state using another communication infrastructure, detailed later in the text (*cf.* §5.1.5).

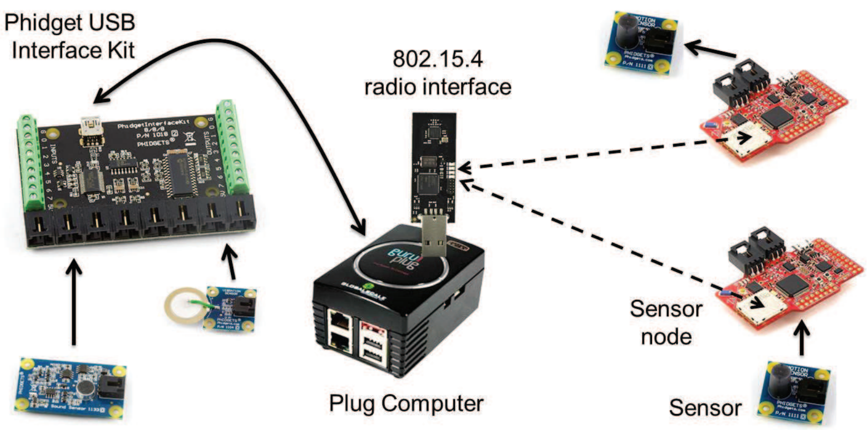


Figure 5.1: Deployment of the Sensing layer

To detect the occupation of a room, we chose the following kinds of sensor: motion, sound and vibration. In our deployment, both the kitchen and the living room are equipped with such sensors. Note that those technologies are chosen to guarantee an acceptable level of privacy preservation, easy deployment, low cost and low energy

[1]http://www.zolertia.com/, accessed 8 February 2012
[2]http://www.phidgets.com/, accessed 8 February 2012
[3]http://www.plugcomputer.org/, accessed 8 February 2012
[4]http://www.atmel.com/tools/RZUSBSTICK.aspx, accessed 11 November 2012

consumption. Adopting different technologies may lead to very different considerations and results.

5.1.2 Perception layer

As already illustrated, the Perception layer processes raw sensor data in order to produce Context Attribute values. Taking our example, the Perception layer interprets the three measures of motion, sound level and vibration as evidence of the presence of somebody. It performs data fusion to calculate a reliable estimate of the presence in the room and sends the results to the upper layer. This will interpret them as Context Attribute values, the input of the Context Spaces Theory reasoning.

To perform its task, the Perception layer exploits a Belief Functions theory implementation, called *THE GAME*[5]. Among other features, THE GAME offers tools to build the sets of mass functions, implements different combination rules and decision-making tools.

The Perception layer runs on the Plug Computers, processing the raw data collected from sensors. Each room is equipped with a Plug Computer, which performs fusion of data provided by the sensors located in the same room, thus executing a portion of the Perception layer in a distributed fashion.

5.1.3 Situation and Context Identification Layer

The Situation and Context Identification layer reasons on runtime Context Attribute values to infer the ongoing Situation Spaces. To this aim, the layer uses an implementation of the Context Spaces Theory called *ECSTRA*, which we will illustrate in Section 5.2. The modifications and extensions of ECSTRA that were realized in the context of this doctoral research are presented in Sections 5.3 to 5.5.

The Context Spaces model and reasoning approaches are inherently distributed. A Situation Space refers to a particular set of interesting Context Attributes, so reasoning can be performed by any device possessing updated values of those Context Attributes. We leverage this property to realize a distributed reasoning architecture, where the functionalities are directly provided by augmented appliances, which can reason about situations using the Context Spaces theory.

Augmented appliances adapt their behavior depending on the results of the reasoning process. The Exploitation layer has the role of providing them with such results, as illustrated below.

5.1.4 Exploitation layer

Applications can be notified about the occurrence of situations by leveraging the Exploitation layer. Applications can subscribe to the Situation Spaces they are interested in and be notified as soon as those situations occur. This allows applications to adapt their behavior depending on context and situations.

We use the concept of Situation Space to model the conditions under which a functionality has to be provided. In other words, to provide a service, we first define

[5] *THeory of Evidence in a lanGuage Adapted for Many Embedded systems* (THE GAME)

the situation under which that functionality is suitable. Then, the distributed context-aware entities will reason on the occurrence of that situation and decide the most adapted action to perform. For instance, a context-aware touch-screen may provide users with a contextual interface at the right moment, allowing them to take over a hotplate left unattended (*cf.* §6.1).

The Exploitation layer was developed by implementing additional components of ECSTRA, the Context Spaces Theory implementation that we adopted. We refer to §5.5 for a description of those additional components.

5.1.5 Communication between layers

The layers composing the system have to exchange information. In a first step, the Sensing layer (layer 1) communicates sensor readings to the Perception layer (layer 2). In a second step, the latter layer sends Context Attribute and confidence values to the Situation and Context Identification layer (layer 3). Then, layer 3 sends Situation Spaces' reasoning results to the Exploitation layer (layer 4), which notifies interested applications.

From a deployment perspective, communications must happen between distributed devices. Indeed, Context Attribute values are calculated by Plug Computers and are communicated to the augmented appliances.

The communication between the devices composing the demonstrator was enabled by a wireless IEEE 802.11 (Wi-Fi) network in infrastructure mode. The role of Access Point (AP) was held by one of the Plug Computers, as shown in Figure 5.2.

Figure 5.2: Deployment and communications between devices

As for the application-level mechanisms for the communication between layers and devices composing the system, we refer the reader to Section 5.6. The next section, instead, presents ECSTRA, the adopted Context Spaces Theory implementation.

5.2 ECSTRA — Context Spaces Theory Implementation

The implementation of the Context Spaces theory that we used is called ECSTRA[6] [BZ11b]. ECSTRA is developed and maintained at LTU (Luleå University of Technology), Sweden. ECSTRA is written in Java and licensed under the '3-clause BSD license' (*Berkeley Software Distribution license*). The source code is publicly available[7].

We now describe the aspects of ECSTRA that are required to understand the extensions and applications developed in the context of this work.

5.2.1 Architecture of ECSTRA

The architecture of ECSTRA consists of following components: *Platform Proxies*, *Context Collectors*, *Application Spaces* and *Clients* (see Figure 5.3). We now illustrate the role of each of those components.

Platform Proxy A Platform Proxy is responsible for retrieving the current value of a Context Attribute, realizing the abstraction of the hardware layer. In general, sensors are connected to a platform that retrieves the current readings and provides them to ECSTRA. A Platform Proxy is responsible for collecting sensor values, provided by a specific platform, and providing them to the upper component (the Context Collector) as Context Attribute values.

Since different sensor platforms may coexist, different Platform Proxies are used to realize the abstraction from heterogeneous hardware and provide the upper components with normalized Context Attribute values.

Context Collector A Context Collector is responsible for maintaining a representation of the current state (the Context Sate — see §3.6.2), as aggregation of current values of Context Attributes received through the Platform Proxies. Indeed, new Context Attribute values can be received at different instants. This is due to the fact that real-world events and delivery of data from sensors do not generally occur at the same time. The Context Collector stores the received values and updates them when newer ones become available.

A Context Collector may be responsible for a subset of the Context Attributes. For example, it is possible to define a Context Collector that is responsible for maintaining the current state in a room. This will then retrieve and store the values of Context Attributes related to that room only, such as the occupation of that room. Although other Context Attributes may be available (*e.g.*, those of other rooms), the Context Collector will not maintain them. This can help reducing the amount of information to manage and hence the complexity of processing. The consequence of this modularity is that several Context Collectors can be instantiated simultaneously, each with different

[6] *Enhanced Context Spaces Theory-based Reasoning Application* (ECSTRA)
[7] `http://code.google.com/p/ecstra`, accessed on 8 February 2012

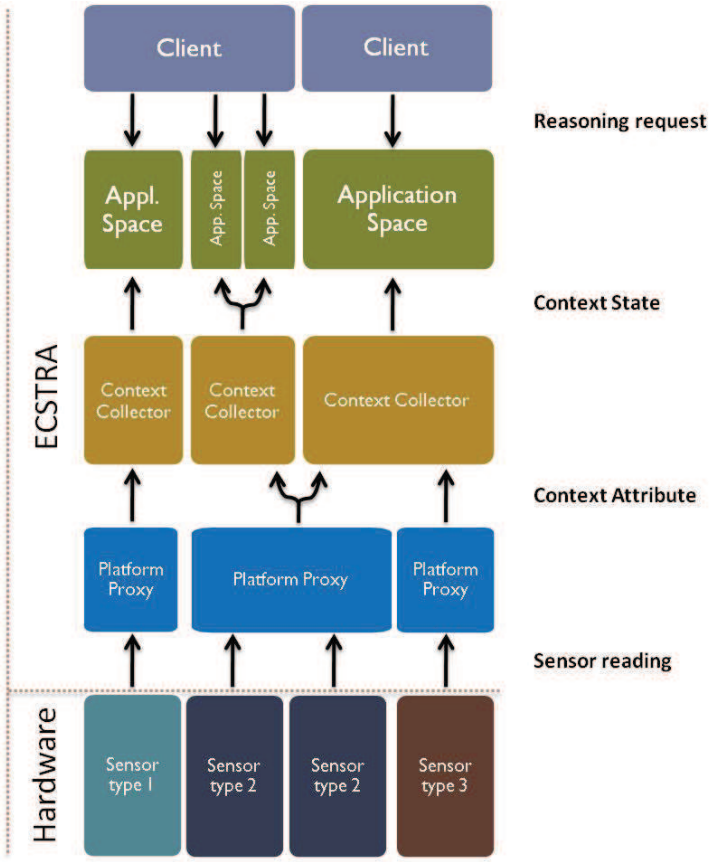


Figure 5.3: The architecture of ECSTRA

responsibilities. In addition, several layers of Context Collector can be superimposed. This allows reusing information by processing only once the Context Attributes values that are interesting for several Context Collectors [BZ11b].

Application Space An Application Space is the place of the reasoning engine, where contextual information is processed to determine the occurrence of situations. To do this, the Application Space must be configured, *i.e.*, the Situation Spaces must be defined within the Application Space. Then, the Application Space can be invoked to perform reasoning about the situations.

An Application Space subscribes to one (and only one) Context Collector, from which it receives the current Context Sate as a collection of each interesting Context Attribute's most recent value. The Application Space can then be invoked to realize the reasoning based on the Context Sate and on the available Situation Space models.

Client A Client represents the "user" of ECSTRA. It subscribes to one or more Application Spaces, to which it submits its requests for reasoning about situations. An example of Client is a context-aware application that uses the capture and inference capabilities of ECSTRA to modify its operation in a context-aware fashion.

To summarize, ECSTRA receives sensor readings, corresponding to Context Attribute values, one by one, through a Platform Proxy. The Platform Proxy then forwards each incoming Context Attribute value to the subscribed Context Collector, which incorporates the new value into the existing Context State, possibly replacing the old value of the Context Attribute. The Context Collector sends the complete Context State to the Application Space, which stores it and can perform reasoning upon request of the Client.

5.2.2 Advantages of ECSTRA

ECSTRA presents several advantages that make it an ideal software tool for realizing the Situation and Context Identification layer of our context-aware system. We now illustrate such advantages.

Distributed Execution

The entire stack of software components of ECSTRA can be run in a distributed fashion on devices and augmented appliances. For this, we can incorporate all the architectural components constituting ECSTRA in an augmented appliance, that is:

1. one or more Platform Proxies, from which the appliance retrieves the values of Context Attributes;
2. one or more Context Collectors, responsible for maintaining all or a subset of the available Context Attribute values;
3. one or more Application Spaces, containing Situation Spaces that require the availability of certain Context Attributes, provided by the Context Collector(s);
4. one or more Clients, used by the appliance to invoke the reasoning about situations.

Such a highly distributed reasoning architecture avoids performance bottlenecks and the issues connected to having a single point of failure.

Modular Architecture

With its modular architecture, ECSTRA perfectly fits for the distributed architecture of our system, where applications and contextual reasoning are executed by resource-constrained devices and appliances.

In particular, the amount of information to be managed by each appliance can be limited by configuring the Context Collectors so that they only keep track of the Context Attributes that are required by the appliance.

The same principle can be applied to reduce the complexity of contextual reasoning performed by each appliance. To this end, one can configure the Application Spaces with the Situation Spaces that are required for the specific appliance only. Furthermore, reasoning can be selectively performed on the interesting situations only.

Despite these advantages, ECSTRA also presents some characteristics that are not well-suited to our context-aware system. Such limitations, which asked for modifying and improving ECSTRA, are illustrated below.

5.2.3 Limitations of ECSTRA

With respect to the needs and requirements of our context-aware system, we identified and solved three classes of shortcomings in ECSTRA, as illustrated below.

Management of uncertainty, imprecision and time The first class of shortcomings directly follows from the limitations of the Context Spaces Theory that we highlighted in the previous chapter. Namely, the unsuitability of the management of uncertainty/ignorance and the lack of tools to model the temporal dimension of situations. We show how we addressed such shortcomings in Sections 5.3 and 5.4.

On-demand reasoning In our context-aware architecture, augmented appliances provide adapted functionalities based on the occurrence of situations. The Exploitation layer is in charge of notifying them as soon as an interesting situation occurs.

With ECSTRA, Clients get notified whenever a new Context State is available and have to explicitly trigger the reasoning by interrogating the Application Space component. In other words, ECSTRA does not include an Exploitation layer.

To make up for this lack, we need to implement the required components to allow Clients to subscribe to interesting situations and to automatically get notified of their occurrence. Section 5.5 presents our choices to implement them.

From sensor data to situations The third class of shortcomings of ECSTRA directly follows from the approach of the Context Spaces Theory, consisting in directly dealing with sensor data. In ECSTRA, indeed, sensor readings are directly used as Context Attribute values.

As we said, our context-aware system, instead, preprocesses sensor data at the Perception layer, which then provides as output the Context Attribute values (*cf.* 4.3.1). Using ECSTRA in our system thus requires modifying the lowest component of its stack: the Platform Proxy. We illustrate a solution to this issue in Section 5.6.

This section described ECSTRA, as well as its advantages and shortcomings. As already announced, the next two sections address the first class of shortcomings of ECSTRA. Namely, the next section shows how to handle uncertainty, imprecision and ignorance with ECSTRA. Then, Section 5.4 will present how to model the temporal dimension.

5.3 Handling Uncertainty and Ignorance with ECSTRA

ECSTRA was extended with new functionalities reflecting the modifications to the Context Spaces Theory that are proposed by this doctoral research. This was partially achieved by collaborating with Andrey Boytsov, the designer of ECSTRA at Luleå

University of Technology. We now illustrate the implementation choices for the new functionalities.

5.3.1 Reconsidering the management of uncertainty

As explained in §4.4, this doctoral research proposed a new way to take into account the uncertainty of Context Attribute values, as provided by the Perception layer of the architecture.

Remember from the description of the Context Spaces Theory that the confidence in a situation is calculated through a matching function μ, which includes several heuristics to handle different aspects of real-world situations. Our contribution was that of proposing a new heuristic that takes into account the uncertainty of Context Attribute values.

ECSTRA implements the matching function originally proposed by the Context Spaces Theory, that is:

$$\mu\left(C^t, S, \sum\right) = \sum_{i=1}^{n} w_i \kappa_i(a_i^t) \tag{3.6.1}$$

The new matching function, proposed in this dissertation, is:

$$\mu(C^t, S, \sum, \Delta) = \sum_{i=1}^{n} w_i \kappa_i(a_i^t)\delta_i^t, \tag{3.6.1}$$

The implementation of the new matching function is straighforward, given that the runtime confidence of Context Attribute i, δ_i^t, is provided by the Perception layer. In case such confidence value is not provided, the default value is equal to 1.

5.3.2 Considering Imprecision and Ignorance

As explained in §4.5, we introduced a solution to explicitly consider imprecise Context Attribute values and ignorance when modeling and reasoning about situations. This is required in order to propagate imprecise information and ignorance from the Perception layer to the Situation and Context Identification Layer.

We opted for a straightforward representation of imprecise values and ignorance. For this, we proposed to define additional values for each Context Attribute. The new set of possible values for a Context Attribute is the collection of all the possible subsets of the original one (*cf.* §4.5).

This allows to reuse the full expressive power of existing modeling and reasoning tools. For this reason, no modifications to ECSTRA were required to handle imprecision and ignorance.

Having described the new solutions for handling uncertainty, imprecision and ignorance, we can now proceed with the introduction of the temporal dimension.

5.4 Modeling the Temporal Dimension of Situations

As explained in Section 4.6, some of the targeted functionalities require considering the temporal developments of situations. For instance, knowing how much time a person has spent in a room helps managing the heating system. For this reason, we need to also consider the temporal dimension in the definition of Situation Spaces.

ECSTRA does not natively provide any tool to model time constraints. Whenever a new Context Attribute is available through a Platform Proxy, the new value is communicated to the upper components. No timestamp is attached to Context Attribute values and no history of previous values is maintained.

For this reason, we designed new modeling tools to integrate the temporal dimension in ECSTRA. This section illustrates such tools.

5.4.1 Realizing a temporal dimension with ECSTRA

To add the temporal dimension to ECSTRA, we developed three new components: the *History Keepers*, the *Time-Adding Context Collector* and the *Synchronizing Context Collector*. Such tools are assembled and integrated into ECSTRA as depicted in Figure 5.4. We now provide the details of such design choices.

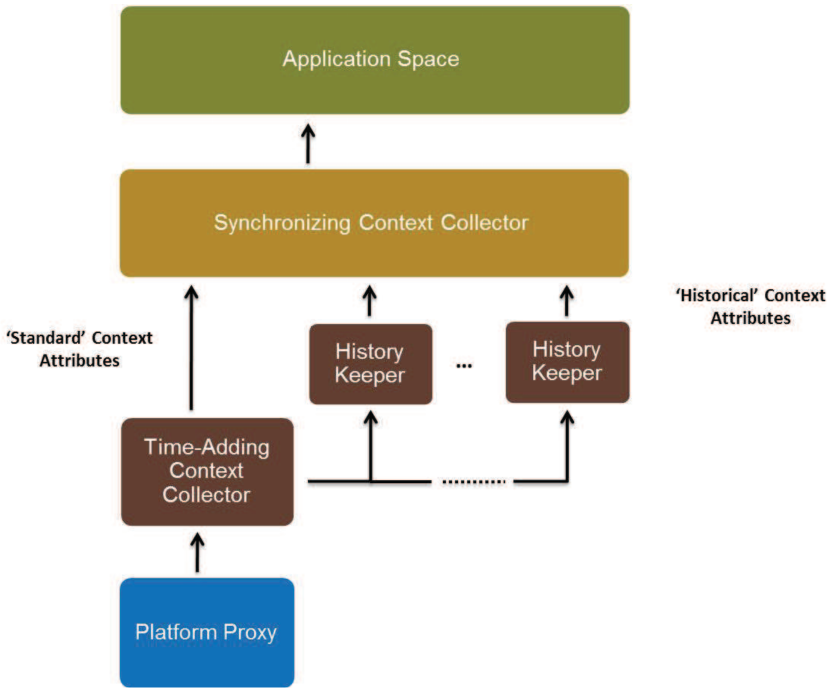


Figure 5.4: New components implementing the temporal dimension

History Keepers A History Keeper generates new Context Attributes that keep track of a particular property of the history of some monitored Context Attributes' values.

Several different history keepers were developed. Examples are the *Averaging History Keeper*, and the *Seeking History Keeper*. The Averaging History Keeper returns the mode of a Context Attribute over time, *i.e.*, the value that appears more frequently, over a sliding window. The Seeking History Keeper returns the number of milliseconds that elapsed since when a certain value of the monitored Context Attribute was observed. For instance, a Seeking History Keeper can publish a Context Attribute that indicates how much time elapsed since when the presence of somebody was detected in a room.

Time-Adding Context Collector Given that, in ECSTRA, no timestamp is attached to Context Attribute values, we developed an additional component: the Time-Adding Context Collector.

A Time-Adding Context Collector incorporates in the Context State an additional Context Attribute that specifies the current time, in milliseconds. It replaces the original Context Collector, so that each time that a Context Attribute is received through the Platform Proxy, the Context Attribute is associated with a timestamp.

Such timestamp allows the History Keepers to correctly build the history of values. For instance, the previously cited Seeking History Keeper will use the timestamp of Context Attributes to count the time elapsed since when the presence in a room was observed.

Synchronizing Context Collector A third-layer Context Collector is necessary to reassemble both the 'standard' Context Attributes and the 'historical' ones, before forwarding them to the Application Space, as shown in Figure 5.4.

To this end, the *Synchronizing Context Collector* waits, before forwarding the Context State, that all the Context Attributes derived from the same 'real' Context Attribute are received. To this end, it checks whether a new timestamp is received before forwarding the Context State (a new time stamp is received only when a Context Attribute has transited through the Time-Adding Context Collector).

This guarantees that, whenever a new Context State is delivered to the Application Space, such Context State contains the latest value of each 'real' Context Attribute and of all the corresponding Context Attributes generated by the History Keepers.

For instance, both the Context Attribute indicating the presence in a room and the Context Attribute indicating the time elapsed since the last detection of presence in that same room will be delivered together. This allows avoiding the case in which a presence is detected but the historical Context Attribute states that the last presence was observed long time ago.

5.4.2 Benefits of the temporal dimension

The new tools allow to take into account the history of Context Attribute values in the model of a situation. For instance, the concepts of presence and absence can be extended over time using such tools.

Considering the example of a person leaving a room and then quickly returning, using the newly introduced tools we can prevent the system from reacting too quickly to the initial exit. We can also model a situation so that it is triggered only after that a person has spent a certain amount of time inside or outside the room. We refer the reader to Chapter 6 (and in particular §6.3) for a detailed illustration of the use and benefits of the new tools through an example.

This section illustrated how to add the management of the temporal dimension to ECSTRA. The next section introduces the implementation choices that we made to realize the Exploitation layer, allowing applications to be notified about the occurrence of interesting situations.

5.5 Realizing the Exploitation layer

As explained in Section 5.1.4, applications can be notified about the occurrence of situations by leveraging the Exploitation layer. Applications can subscribe to the Situation Spaces they are interested in and be notified as soon as those situations occur. This allows applications to adapt their behavior depending on context and situations.

To realize the subscription to the interesting Situation Spaces, ECSTRA has been extended with additional components called *Context Manager* and *Notifier*.

Notifier A Notifier waits for a new Context State to be available and, when this happens, it triggers the reasoning about interesting situations on the Application Space. Then, it returns the situation reasoning result to the Context Manager.

Context Manager The Context Manager is used by Clients to subscribe to interesting situations (or logical expressions involving Situation Spaces — *cf.* §3.6.4). Whenever the confidence in the occurrence of a situation changes, the Context Manager communicates the new value to Clients.

This section illustrated the new components that were developed to realize the Exploitation layer. These components are depicted in Figure 5.5. The next section illustrates how ECSTRA communicates with the other layers of our context-aware system. To this end, we first present the general communication paradigm and then illustrate how ECSTRA was modified to adopt it.

5.6 Exchanging Information between Layers

We now illustrate how ECSTRA is connected to the Perception layer, in order to receive the Context Attribute values, used to infer the occurrence of situations.

We first describe the requirements of the communication mechanism to design. Such requirements follow from our previous choices in terms of implementation and deployment. Based on such requirements, we select a communication paradigm, the Publish/Subscribe, and present the adopted implementation. Finally, we describe the additional components that allow ECSTRA to communicate using such paradigm.

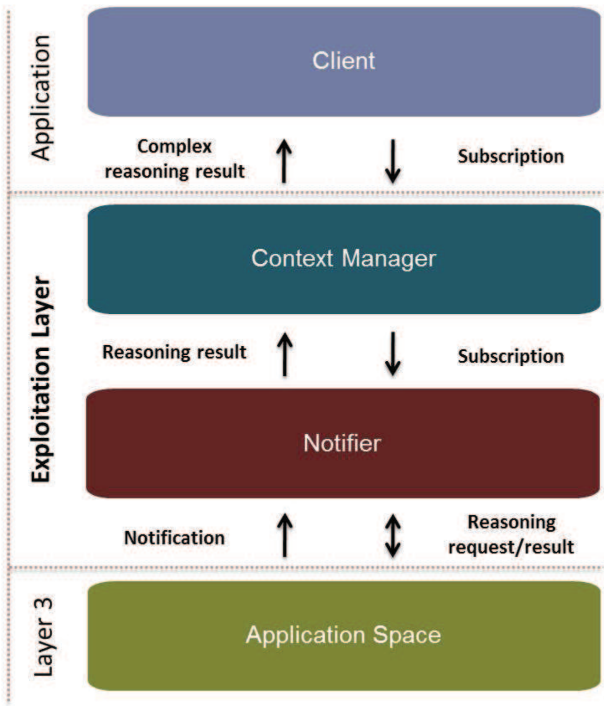


Figure 5.5: Architecture of the Exploitation layer

5.6.1 Requirements of the Communication Mechanism

Communications between layer 2 and layer 3 of the architecture are mainly focused on the exchange of Context Attribute values. As previously detailed, the Context Attribute values can be obtained by layer 2 (which performs sensor data fusion) or directly provided by appliances. The couple consisting of a Context Attribute value and the respective confidence must be communicated to layer 3 as input to the Context Spaces theory.

Considering the distributed nature of the system under development, it is conceivable that even in the simplest scenario, layer 2 is deployed on a different device than the one running layer 3. It should be possible to communicate the Context Attribute values to another device through a communication medium. In addition, since layer 3 can be performed in a distributed manner by multiple devices, a Context Attribute produced by an entity must be sent to multiple recipients.

These requirements ask for the establishment of a point-multipoint communication mechanism, capable of delivering messages while being transparent to the physical configuration of the system. For instance, layer 2 should be able to provide as output the Context Attribute values without the need to know which functional components and devices will use this information.

In the remainder of this section, we describe a communication model that meets these criteria. Figure 5.6 illustrates what we aim to obtain as a result.

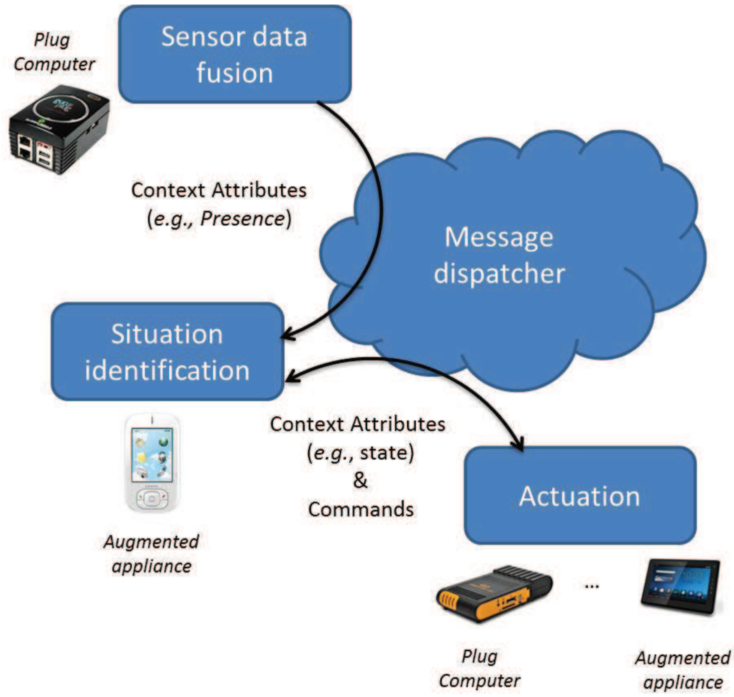


Figure 5.6: Exchanging information between layers and devices

5.6.2 Elvin Publish/Subscribe paradigm

The *Publish/Subscribe* paradigm can meet the identified communication requirements. It is a communication model characterized by a decoupling between sender and recipient. The exchanges are centered on the notifications, which serve both as content of the communication (the message) and as mechanisms for addressing and filtering the notifications themselves.

Elvin open specifications [ELV] provide a routing service that uses a distributed Publish/Subscribe paradigm. A notification message is transmitted and consists of one or more fields characterized by a name and a value, which are also used to filter and route the notification. Using Elvin to communicate the value of a Context Attribute, the notification may contain for example the field: *(Presence, TRUE)*, where *Presence* is the name of the Context Attribute and *TRUE* is its value.

Using a Publish/Subscribe paradigm *à la* Elvin, any client can subscribe to notifications through the *Elvin Subscription Language*[8]. This language allows clients to specify the notifications they wish to receive based on their content. In our example, a client may declare that it is only interested in the Context Attribute values *Presence*. The subscription language allows specifying other constraints on the content of the notifications to be delivered; regular expressions are also supported.

[8] Elvin Subscription Language, http://avis.sourceforge.net/subscription_language.html, Accessed on 25 November 2011

5.6.3 Benefits of Elvin

The main advantages of the Elvin Publish / Subscribe approach in the case of our system are:

- The ability for any device to subscribe to interesting context information and to receive such information as it becomes available, without the need for further configurations or knowledge of the system topology.

- The simplicity and scalability of the approach, where routing and filtering of notifications is entirely performed by Elvin router(s), which simplifies the task of producing and consuming contextual information.

5.6.4 Adopting Elvin

We adopted Avis[9] as an open-source implementation of Elvin specifications.

Avis consists of two main components: the *router* and the *client*. The router manages connections and client subscriptions, performing routing and filtering of notifications. It is implemented in Java and is compatible with any platform that supports Java version 5 or higher. The client connects to the router and can publish and/or subscribe to notifications. The client library is available in C and Java languages.

Avis is used in our system for publishing Context Attribute values and confidence. To do this, layer 2 instantiates a client (in the current implementation of layer 2, based on the Avis client library for C language). Meanwhile, layer 3 instantiates a Java client and subscribes to notifications that contain Context Attribute values.

When a new value of a Context Attribute is available (either as a result of data fusion or following a change of state of an augmented appliance), layer 2 publishes the Context Attribute value and the corresponding confidence. If the Elvin client instantiated by layer 3 has subscribed to updates of the Context Attribute (or, simply, if it has subscribed to updates of all Context Attributes), it receives the notification. The new value of Context Attribute is processed by layer 3, *i.e.*, reasoning is performed to detect ongoing situations.

When layer 3 is implemented in a distributed fashion by several devices, each of them will instantiate an Elvin client and subscribe to interesting Context Attributes. This process is depicted in Figure 5.7.

The advantage of adopting Elvin in this scenario is the simplicity of the task performed by the device, which does not have to filter uninteresting notifications. This operation, in fact, is executed by the Elvin router, using the subscription issued by the device using Elvin Subscription Language. This also reduces the number of messages exchanged between the devices, having positive effects on the system load and energy consumption.

Avis is also used for other scopes in our architecture. Namely, we also used Avis to send commands to appliances and to allow appliances to publish Context Attributes concerning their state. This achieves the goal that was depicted in Figure 5.6.

[9] Avis Event Router, http://avis.sourceforge.net/, Accessed on 25 September 2011

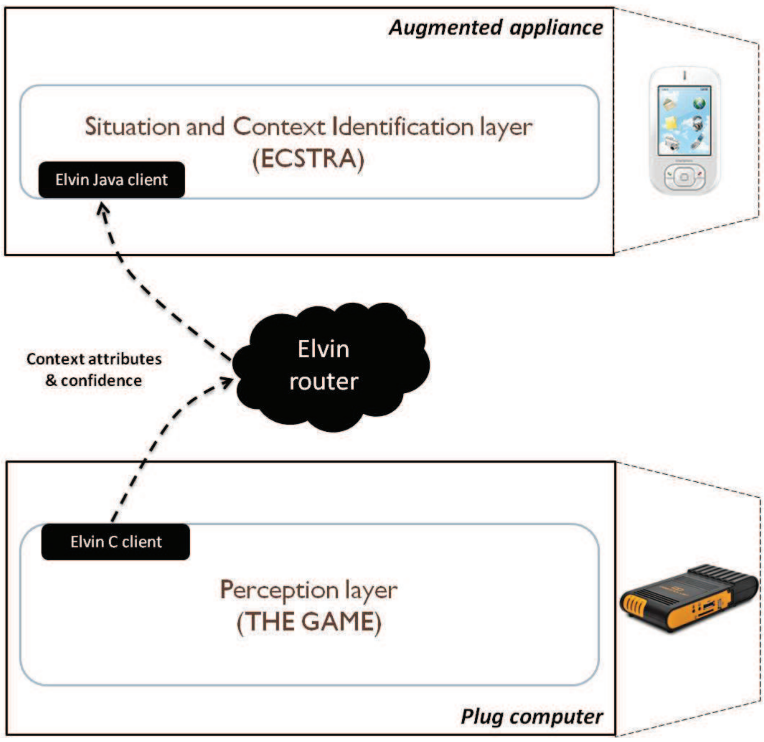


Figure 5.7: Context Attribute exchange using Elvin

5.6.5 Empowering ECSTRA to Use Elvin

In ECSTRA, Context Attribute values are produced by Platform Proxies, which have the role of abstracting from the different hardware implementations of the sensors (*cf.* §5.2).

In our architecture, sensor data are pre-processed by the Perception layer, which directly provides Context Attribute values using Elvin. Thus, ECSTRA has to be extended with a mechanism for receiving Context Attributes values as Elvin notifications. In the following paragraphs, we go into details of this mechanism.

Connection Manager In order for ECSTRA to receive Context Attribute values as Avis notifications, we designed a new component called *Connection Manager*. The Connection Manager receives notifications from a Publish/Subscribe middleware and delivers them to subscribed receivers.

The Connection Manager is a generic component for exchanging information using the Publish/Subscribe paradigm. A specialized Connection Manager has been implemented to handle the connection to an Avis router, sending and dispatching notifications. Other kinds of Connection Managers may be implemented to handle the connection to other Publish/Subscribe middleware.

The Connection Manager is not only used to exchange Context Attribute values. Since we also use Avis to send commands to appliances, the Connection Manager provides the appliances with an additional layer of abstraction from the communication mechanism. Other uses of the Connection Manager might include the remote dispatching of reasoning results.

Notification Processor A component called *Notification Processor* has been implemented, providing an abstraction from the different kinds of notifications. Notifications may differ in their format, specific to the adopted communication mechanism, and in their syntax, specific to the particular kind of transmitted information.

There exists a different Notification Processor for each kind of information to transmit and each kind of middleware. For instance, there is a Notification Processor to handle Context Attributes delivered through Avis.

The Notification Processor is used by a Connection Manager to handle any kind of notification. A specific Connection Manager, like the Avis Connection Manager, will only deal with notifications received through Avis. However, it will be able to process any kind of information received through Avis, leveraging the Notification Processor. So, the Connection Manager will be able to receive and process Context Attribute values, appliance commands and any other kind of notification developed in the future.

Remote Platform Proxy A special kind of Platform Proxy has also been implemented, called *Remote Platform Proxy*. It subscribes to one or more Connection Managers and is notified whenever a new Context Attribute is received through the communication mechanism handled by the Connection Manager, *e.g.*, Avis.

In other words, this new component is a Platform Proxy that, instead of abstracting from the hardware details (as do the other existing Platform Proxies) abstracts from the underlying communication system details.

Obviously, the Remote Platform Proxy can be used at the same time than other Platform Proxies that abstract from sensors that are physically connected to the same device. This preserves the modular architecture of ECSTRA.

An overview of the communication components added to ECSTRA is depicted in Figure 5.8.

The overall architecture, resulting from the modifications that were realized to ECSTRA, is presented in Figure 5.9.

Starting from the bottom, the figure shows that augmented appliances and the Perception layer can communicate Context Attribute values to ECSTRA through a Publish/Subscribe middleware.

Context Attribute values are then timestamped by a Time-Adding Context Collector (TACC) and the Context State is enriched with a temporal dimension by the History Keepers (HK). A Synchronizing Context Collector aggregates the resulting Context Attributes and provides them to the Application Space.

Reasoning is invoked on the latter component by a Notifier, which provides the reasoning results to the Context Manager. This component calculates the occurrence of complex situations, *i.e.*, logic expressions involving situations, and provides the result to the Client, which had declared its interest through a subscription.

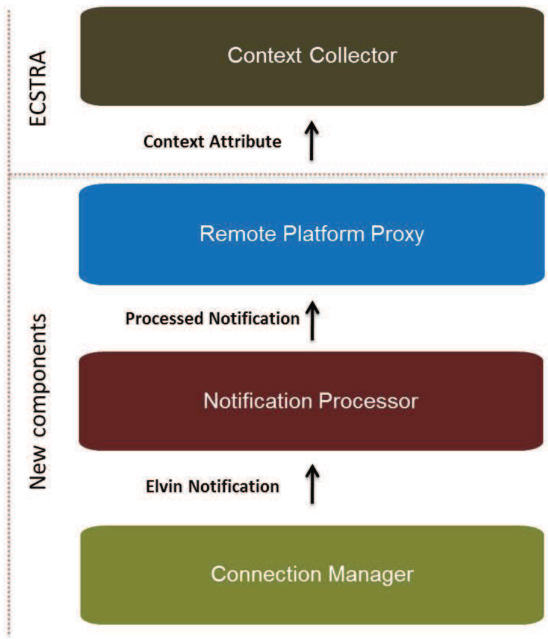


Figure 5.8: New components using Elvin

5.7 Conclusion

This chapter presented the implementation and deployment of the proposed context-aware system.

We first illustrated an overview of the deployment choices. In particular, we showed that our system leverages lightweight instrumentation made of communicating sensor nodes, small computers and household appliances. Such technologies guarantee minimum change to the appearance of the house and to users' behavior. They also minimize privacy issues and, thus, the acceptability of the overall system.

Appliances are augmented with the capability of reasoning about situations, provided by ECSTRA, an implementation of the Context Spaces Theory. This approach realizes a highly distributed reasoning architecture, avoiding performance-bottleneck and single-point-of-failure issues. The next chapter will evaluate the reasoning performance of such architecture (*cf.* 6.5).

This chapter also described the improvements that we made on ECSTRA. We developed the management of uncertainty, imprecision and ignorance, as well as the tools required to add a temporal dimension to the process of reasoning about situations. When realizing such improvements, we reused to the maximum extent the existing modeling and reasoning tools of the Context Spaces Theory.

We also designed and implemented the Exploitation layer, needed to notify applications about the occurrence of interesting situations. Then, we enabled the communication between the layers of the context-aware architecture, using a Publish/Subscribe middleware. This supports the communication between the heterogeneous and distributed components of our system. Finally, we designed and implemented new generic software components that allow ECSTRA to take part in a Publish/Subscribe information flow.

The next chapter will evaluate the presented context-aware system by describing an application that was deployed using it. Such application exploits the tools that we described in this Chapter and was designed following the interdisciplinary approach illustrated in Chapter 2.

In addition to presenting the application, we will also evaluate the behavior of the system when facing a realistic domestic scenario. To this end, we will show how to use the available modeling tools to address the imperfections of the sensing and reasoning architecture.

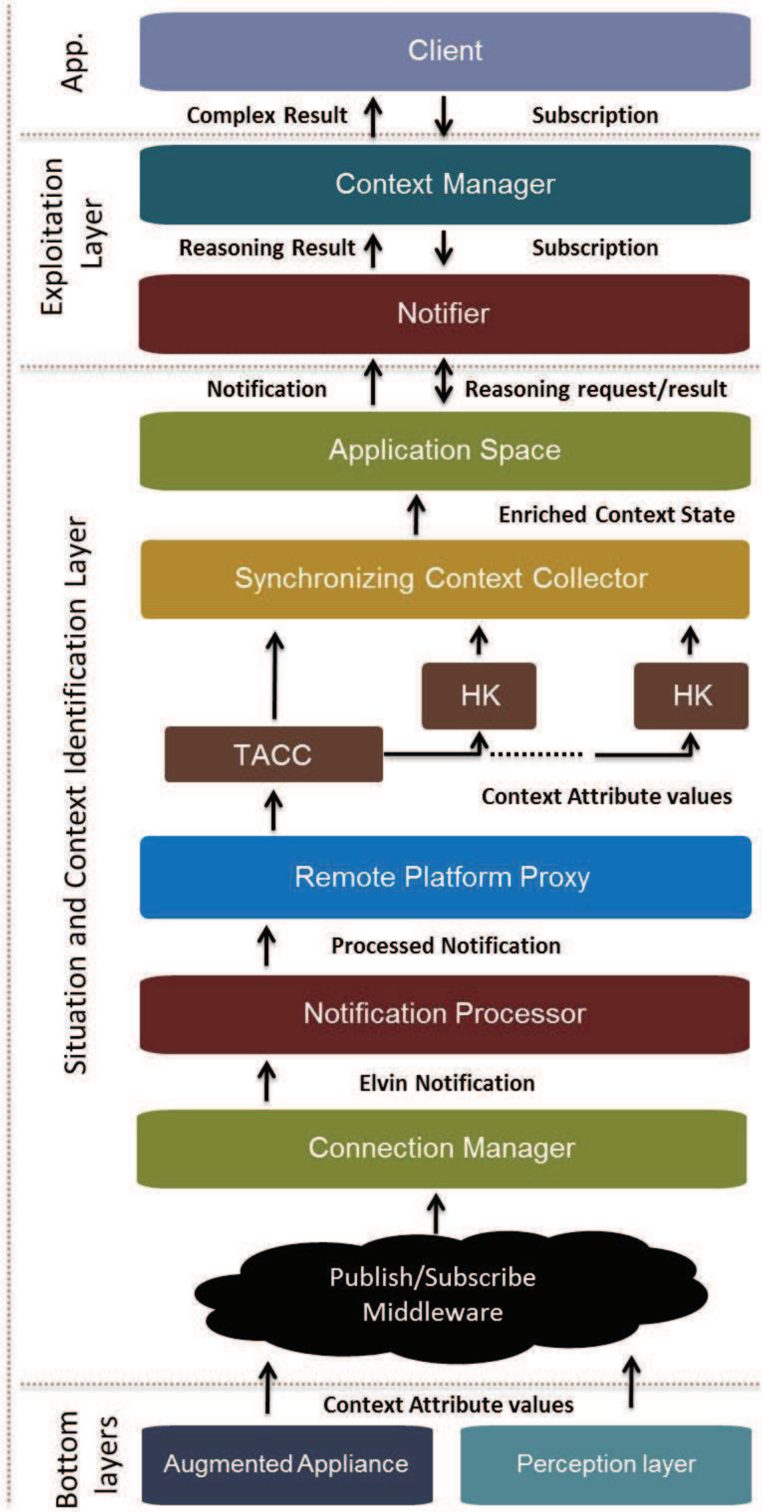


Figure 5.9: The new architecture of ECSTRA

Chapter 6

Evaluation Through an Application

The previous chapter illustrated our choices for the implementation and deployment of a Smart Home prototype. We now validate such choices.

To this end, we first illustrate an application that we realized, in Section 6.1. Such application leverages our context-aware system to improve the sustainability of a real-world domestic situation.

We also describe a scenario that challenges the context-aware capabilities of the system. Then, we show how the tools implemented in the context of this doctoral research can be used to address that issue (Sections 6.2 to 6.4).

Finally, we demonstrate that our context-aware system can be adopted in real-world conditions. For this, Section 6.5 shows that the system can be used to reason about complex situations in a distributed fashion, being deployed on resource-constrained devices.

6.1 *Cooking Assistant* Application

We now illustrate an application that was realized leveraging the interdisciplinary design approach presented in Chapter 2. We aim at creating a *Cooking Assistant* application, which realizes a sustainable situation described in the scenario, which we remind below (*cf.* paragraph 4 in the scenario of Section 2.4.3):

> The stovetop heating elements in the kitchen are activated by a person. Nothing happens as long as activity remains in the kitchen. In contrast, if an extended absence is detected in this room (in our scenario, the two people are now in the living room), the option of turning off the stovetop is offered to the user through a tablet in the living room.

We describe the application as follows: first, the motivation underlying the application is presented, together with a description of the functionalities to be provided to users. Then, we build the computational models needed to realize the context-aware functionalities, step by step.

The focus is on the models for the Situation and Context Identification layer, as this layer constitutes the main technical contribution of this doctoral research. To this end, we present our modeling choices using the original Context Spaces Theory. We will then present a challenging scenario and show that the proposed improvements of the Context Spaces Theory allow solving the emerging issues.

6.1.1 Why a Cooking Assistant?

Cooking is an engagement often characterized by the use of several energy-intensive appliances. These appliances often operate autonomously for the majority of the time, only asking for few interactions with users. For instance, ovens and hotplates are typically turned on and left running until cooking ends. In Chapter 2, we called this a *procedural need for manual intervention* (*cf.* §2.3.2).

Experience shows that cooking appliances can be forgotten, resulting in waste of energy and risks for occupants' safety. In particular, many oversights happen when the appliance is left running and the person moves to another room.

Unfortunately, automatically turning off the appliance is not advisable. As we highlighted in Chapter 2, domestic activity is so dynamic and articulated that distinguishing whether the appliance has been forgotten or purposefully left on (for instance, for the meal to simmer) is often impossible (*cf.* §2.3).

As described in our scenario, reminders and interaction modalities allowing the remote operation of cooking appliances are better alternatives. Situations suggesting that an appliance might have been forgotten can be modeled and recognized. Whenever one of those situations occurs, a notification can remind the occupant about the appliance and, possibly, propose adapted and local interaction modalities to turn it off.

6.1.2 Description of the Application

The *Cooking Assistant* application recognizes a situation in which the hotplate has been left on when nobody is in the kitchen. It then localizes the inhabitants and proposes them to remotely switch off the hotplate.

We choose the following technologies to realize the functionality: an augmented hotplate, located in the kitchen, which can publish a Context Attribute indicating its state and be remotely operated, and a touch-screen located in the living room, adjacent to the kitchen.

The touch-screen is the reasoning entity and provider of the functionality. It performs reasoning and, when it detects the occurrence of the interesting situation, it displays a dialog box proposing to switch off the hotplate.

6.1.3 Building the Computational Models

Let us call *Follow cooking* the Situation Space whose occurrence causes the service (proposal of remote operation of the hotplate) to be provided. *Follow cooking* is defined as: nobody is in the kitchen, someone is in the living room and the hotplate is on.

Context Attributes

The following Context Attributes are relevant for the application:

- *Presence in the kitchen*, whose value at time t is denoted as a_k^t.
- *State of the hotplate*, whose value at time t is denoted as a_h^t.
- *Presence in this room*, whose value at time t is denoted as a_p^t.

The *Presence in this room* Context Attribute refers to the presence in the same room of the device executing the reasoning. In our case, the touch-screen executing the application will consider the presence in the living room.

For our deployment, we hard-coded the location as a property of the touch-screen. This choice is motivated by the assumption that a touch-screen has a fixed location, to be configured when deploying the Smart Home system. Future homes might indeed be equipped with multi-modal interfaces in every room.

Of course, in case the dynamic localization of the touch-screen is available at runtime (*e.g.*, as in [PTA06]), this information can be used to consider the presence in the relevant room when performing the Context Spaces reasoning. To this end, it is possible to use the distributed reasoning extensions provided by ECSTRA (namely, the context-aware data retrieval [BZ11b]).

Heuristics

The Situation Space *Follow cooking* is denoted as S_{FC}. Remember that, using the original Context Spaces Theory, the confidence in its occurrence is obtained using Equation 3.6.1:

$$\mu(C^t, S_{FC}, \sum) = \sum_{i=1}^{n} w_i \kappa_i(a_i^t)$$

$$= w_p \kappa_p(a_p^t) + w_h \kappa_h(a_h^t) + w_k \kappa_k(a_k^t)$$

To model the Situation Space, we have to define the relevance function and the contribution functions. That is, we have to assign weights to the Context Attributes and assign a contribution to every Context Attribute value. This means fixing w_p, w_h and w_k and defining the functions $\kappa_p(a_p^t)$, $\kappa_h(a_h^t)$ and $\kappa_k(a_k^t)$.

Contribution functions Let us start by defining the contribution functions for the Context Attributes. This means that, for each Context Attribute, we have to fix the contribution that every *value* of the Context Attribute brings to the Situation Space.

For instance, the possible values of the Context Attribute *State of the hotplate* are $\{ON\}$ and $\{OFF\}$ and we have to define a contribution function that specifies how the Situation Space confidence is impacted when the Context Attribute takes each of those values.

When defining the contribution function for a Context Attribute, we want to guarantee that the final confidence in the situation will be impacted as much as possible by the runtime value of the Context Attribute. It is then the role of the relevance function (cf. §3.6.3) to weigh the relative importance of each Context Attribute for the situation.

Remember from §3.6.3 that the codomain of the contribution function is $[0, 1]$. We set to 0 the contribution of the Context Attribute value when it should definitely reduce the confidence in the occurrence of the situation. Conversely, we set the contribution to 1 when the Context Attribute value should definitely increase that confidence. Intermediate contribution values can be specified for Context Attribute values that partially support the situation, although this does not apply to our binary case.

Following the specified empirical method, we define the contribution function for the *State of the hotplate* Context Attribute as follows:

$$\kappa_h(a_h^t) = \begin{cases} 1 & \text{if } a_h^t \text{ is } \{ON\} \\ 0 & \text{if } a_h^t \text{ is } \{OFF\} \end{cases}$$

Following similar considerations, the contribution functions of the *Presence in the kitchen* and *Presence in the living room* Context Attributes are defined as follows:

$$\kappa_k(a_k^t) = \begin{cases} 0 & \text{if } a_k^t \text{ is } \{yes\} \\ 1 & \text{if } a_k^t \text{ is } \{no\} \end{cases}$$

$$\kappa_p(a_p^t) = \begin{cases} 1 & \text{if } a_p^t \text{ is } \{yes\} \\ 0 & \text{if } a_p^t \text{ is } \{no\} \end{cases}$$

Relevance function Let us now define the relevance function. It assigns a weight to every Context Attribute, specifying the relative importance of the Context Attribute with respect to the other Context Attributes that compose the situation.

In our case, the weights of the state of the hotplate and presence in the kitchen should be higher than the other weight because the situation is interesting only if the hotplate is on and nobody is near it.

For this reason and remembering that $\sum_{i=1}^{n} w_i = 1$, we assign the weights in the following way:

$$\left\{ \begin{array}{l} w_h = w_k = 0.4 \\ w_p = 0.2 \end{array} \right\}$$

Decision Making

We defined the Situation Space by specifying the contribution functions and the relevance function. In order to provide the functionality, a decision about the occurrence of the situation *Follow cooking* must be made. Remember from §3.6.4 the criterion for decision making:

$$\gamma = \left(\mu(C^t, S, \sum) \geq \varepsilon\right) \tag{3.6.2}$$

For the decision to be correct, we have to set the value of ε, considering the previously specified parameters of the Situation Space. The Situation Space should be triggered when and only when the Context State, *i.e.*, the collection of runtime values of the Context Attributes, is:

$$C^t = \left\{ \begin{array}{l} a_p^t = \{yes\} \\ a_k^t = \{no\} \\ a_h^t = \{ON\} \end{array} \right\}$$

Given our previous modeling choices, the confidence in the Situation Space in this case is:

$$\mu(C^t, S, \sum) = \sum_{i=1}^{n} w_i \kappa_i(a_i^t) = 0.2 + 0.4 + 0.4 = 1$$

We can set $\varepsilon = 1$, so as to guarantee that the situation will be triggered only when the Context Attributes have the specified values.

6.1.4 Executing the Cooking Assistant Application

After having shown how to build the computational models that allow to recognize the *Follow cooking* situation, we now illustrate the runtime functioning of the context-aware application, depicted in Figure 6.1.

First of all, the Sensing layer measures the physical properties of the environment (sound, vibration and movement in each room). The Perception layer fuses such raw data and establishes a value for the Context Attributes *Presence in the kitchen* and *Presence in this room*. At the same time, the augmented hotplate publishes *State of the hotplate* Context Attribute's value.

The Situation and Context Identification layer receives the values of presence in the rooms and state of the hotplate and maps them to a point in the multi-dimensional space. Whenever the point highlighted in Figure 6.1 is reached, the situation is triggered.

When the *Follow cooking* Situation Space is triggered, the Exploitation layer notifies the application running on the living room touch-screen. This will show a dialog box, proposing to remotely operate the hotplate.

6.1.5 Description of a Challenging Scenario

We described the design and realization of the *Cooking Assistant* application. We now present a realistic scenario and observe how such application is challenged by the uncertainty of the information acquired at the Sensing layer and propagated up to the Situation and Context Identification layer. The following sections will then present how additional tools can be used to evaluate and minimize the risk of providing inappropriate functionalities to inhabitants.

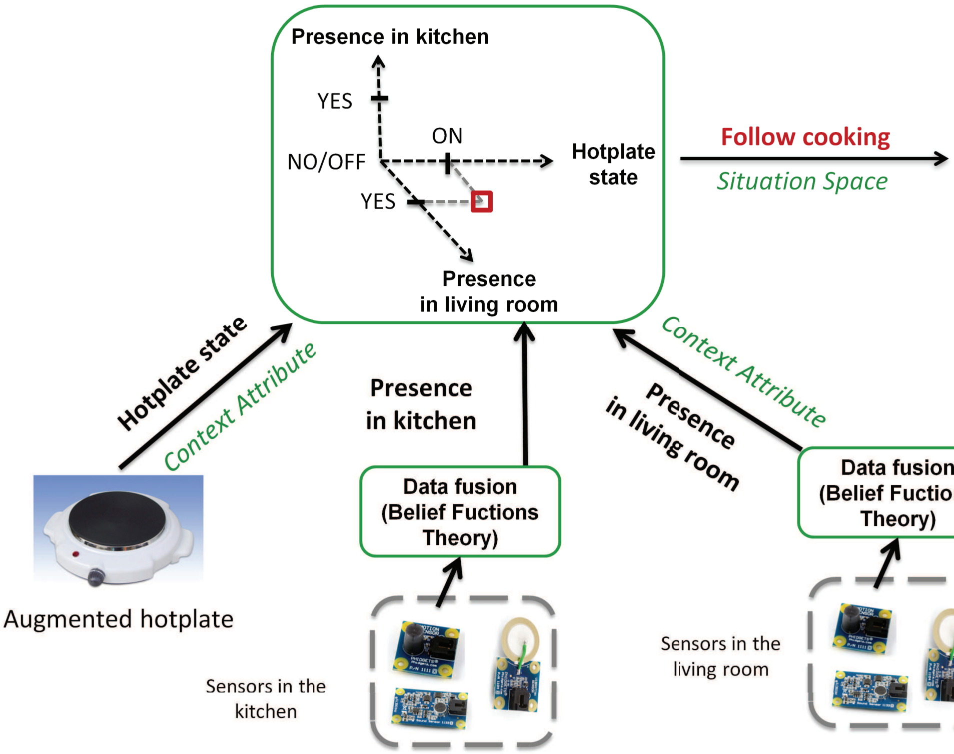


Figure 6.1: The recognition of the situation *Follow cooking*

> Two persons are at home, busy cleaning the house. One person is initially in the kitchen, cooking using the hotplate, while the other one is doing the dust in the living room.
> The system correctly detects the presence of both, due to the remarkable motion, vibration and sound produced. The values of the Context Attributes corresponding to the presence in the kitchen and living room are both set to $\{yes\}$.
> Then, the person in the kitchen takes a pause and seats quietly to read a recipe from a magazine. The Sensing layer no longer detects remarkable motion, sound level or vibrations.
> The Perception layer, fusing sensor data provided by the Sensing layer, is not able to state whether somebody is in the room or not. Thus, the Situation and Context Identification layer is unable to decide if the *Follow cooking* situation is ongoing or not. The system faces a challenging scenario.

The scenario shows that the dependence of the system on sensor data fusion can sometimes put it in a no-win situation. To obtain trustworthy outputs, the data fusion model should be finely tuned for the particular sensor characteristics, locations and environmental conditions specific to the deployment.

This is often not possible to realize for practical reasons: the limited time available for the deployment, the simplicity that has to be guaranteed to realize the targeted plug-and-play deployment of the system, *etc.* Furthermore, very sophisticated, precise and costly sensors should be available, while the sensors described in this scenario have limitations, due to the technological constraints that we have chosen (cost, privacy preservation, *etc.*).

If the Perception layer is unable to take a decision concerning the occupation of the kitchen, the system faces a challenging scenario. The next sections show that solutions are provided by the tools for modeling and propagating information about ignorance and uncertainty, as well as by the temporal aspects of situations.

6.2 Taking Ignorance into Account

In the illustrated scenario, a person is detected in the living room, the hotplate is on and the Perception layer is not sure whether somebody is in the kitchen. Without a way to express this lack of knowledge, the Perception layer (layer 2) is forced to make a decision about the presence or absence of somebody in the kitchen. At layer 3, the Situation Space *Follow cooking* is then triggered or not triggered depending on this decision.

If the decision is wrong (*i.e.*, not corresponding to the real state of occupation of the kitchen), an unsuitable functionality is provided. For instance, the display in the living room shows a notification that turns out to be unnecessary and bothering, as someone is in fact in the kitchen and can manage the hotplate.

To tackle this issue, we need to exploit the tools for modeling and propagating ignorance. These include tools for runtime propagation of imprecise values and ignorance between layers and modeling tools for taking into account that estimation when reasoning about situations.

Imprecision and ignorance are represented by the cardinality of the sets of values sent by the Perception layer to the Situation and Context Identification layer. For example, when neither the confidence on the occupation nor the confidence on the non-occupation of the kitchen are high enough, the Perception layer assigns the imprecise value $\{yes \bigcup no\}$ to the Context Attribute *Presence in kitchen*. This allows the Perception layer to communicate its ignorance to the Situation and Context Identification layer.

Remember that, when modeling a Situation Space, an imprecise Context Attribute value can be assigned a dedicated contribution value (*cf.* §4.5). For instance, concerning the *Follow cooking* Situation Space, the contribution of the value $\{yes \bigcup no\}$ for the Context Attribute *Presence in kitchen* can be set to 0. This means that when the system is not sure of the occupation of the kitchen, it will not trigger the situation, avoiding bothering the inhabitant located in the living room.

Thus, the contribution of *Presence in the kitchen* is redefined as follows:

$$\kappa_k(a_k^t) = \begin{cases} 1 & \text{if } a_k^t \text{ is } \{yes\} \\ 0 & \text{if } a_k^t \text{ is } \{no\} \\ 0 & \text{if } a_k^t \text{ is } \{yes \bigcup no\} \end{cases}$$

Concerning *Presence in the living room*, we may want to display the notification on the touch-screen even if the system is not sure of the occupation of that room. This guarantees that a warning is spread whenever the hotplate has been left unattended. So, for the *Presence in the living room* Context Attribute, the contribution function is redefined as follows:

$$\kappa_p(a_p^t) = \begin{cases} 1 & \text{if } a_p^t \text{ is } \{yes\} \\ 0 & \text{if } a_p^t \text{ is } \{no\} \\ 1 & \text{if } a_p^t \text{ is } \{yes \bigcup no\} \end{cases}$$

6.3 Introducing the Temporal Dimension

We have shown the benefits of the tools for modeling and propagating imprecise information and ignorance, when applied to a scenario for the Cooking Assistant application. We now demonstrate that such tools do not address all the challenges raised by the scenario. Then, we describe how additional help can be provided by the tools for modeling and recognizing the temporal dimension of situations.

6.3.1 Need for a Temporal Dimension

In particular deployments, it might happen that the Perception layer frequently reports an imprecise value for the presence ($\{yes \bigcup no\}$). This might be due to the impossibility of collecting enough evidence to be sure about the emptiness of a room. For instance, in our deployment, the absence of vibration, movement and sound does not prove unequivocally that the room is empty.

With the previous modeling choices, if the Perception layer is unable to decide about the occupation of the kitchen ($a_k^t = \{yes \bigcup no\}$), the functionality is not provided. So, the situation might never be triggered and the functionality never be provided. This means wasting the opportunity of assisting the activity with a suitable service, as the hotplate was in fact left unattended.

6.3.2 Modeling the Temporal Dimension

To tackle this issue, we define an additional Context Attribute, *Long presence in the kitchen*. It is calculated by an Averaging History Keeper (*cf.* §5.4).

The new Context Attribute reports the most frequent value of *Presence in the kitchen* in the last few minutes. Its value at any time t is denoted as a_{LK}^t and is chosen from the same set of values of the original Context Attribute ($\{yes\}$, $\{no\}$ and $\{yes \bigcup no\}$).

The newly defined Context Attribute can be replaced or added to the existing ones in the matching function μ, the equation used to calculate the confidence in the situation (*cf.* §6.1.3).

Let us now define the contribution function of the new Context Attribute. We want to provide the functionality only if the system is certain that the kitchen is unoccupied, but we also want to avoid that ignorance lasting over time prevents the functionality from being triggered when it would be necessary. Thus, we define the contribution of *Long presence in the kitchen* as follows:

$$\kappa_{LK}(a_{LK}^t) = \begin{cases} 0 & \text{if } a_{LK}^t \text{ is } \{yes\} \\ 1 & \text{if } a_{LK}^t \text{ is } \{no\} \\ 1 & \text{if } a_{LK}^t \text{ is } \{yes \bigcup no\} \end{cases}$$

This guarantees that, when nobody is in the kitchen, the system will eventually provide the functionality, even if the Perception layer persists in stating its ignorance. Thus, if the Perception layer has returned $\{yes \bigcup no\}$ for the last few minutes, this is interpreted as evidence of the fact that the kitchen is unoccupied. The ultimate result is that the hotplate will never be left without supervision for long time.

6.3.3 Benefits of the Temporal Dimension

Introducing the temporal dimension improves the recognition for several reasons. The remainder of this section illustrates them.

Noise filtering

Long presence in the kitchen returns the most frequent value of *Presence in the kitchen* over a sliding window. If the presence of a person is overlooked for few instants only, this will not be reported by the newly introduced Context Attribute. Thus, this allows filtering the fluctuations of the output of the Perception layer.

Improved timing

Long presence in the kitchen provides a way to detect that the Perception layer has returned $\{yes \bigcup no\}$ for more than a certain amount of time. This is considered in the reasoning process, so as to make a decision accordingly.

More specifically, if the occupation of the kitchen remains uncertain, the application eventually decides to display the notification on the living-room touch-screen. This avoids the case in which the functionality is never provided. In this sense, the additional Context Attribute can be considered as a "timer" that notifies the application whenever the uncertainty has lasted loo long.

One can draw an analogy with a standard timer used in many domotic systems. Such systems assume that occupants are absent after a fixed amount of time that nobody has been detected. Our system provides an improved solution because we use our timer to know *what* has happened in the last few minutes: it could be detection, absence of detection or ignorance. The application can then act accordingly.

Contextual gap tackling

Introducing the temporal dimension also allows taking the contextual gap into account. More specifically, in Chapter 2 we highlighted that the time elapsed since when a person has entered a room provides evidence to identify the most likely engagement of the person (*cf.* §2.5.1).

In our application, using *Long presence in the kitchen*, the service is provided only if the person has been absent from the kitchen for enough time. This guarantees that users are not bothered when they leave the kitchen for a moment, intending to quickly come back.

To summarize, modeling and reasoning about the temporal dimension improves the integration between layers and the recognition of real-world situations.

6.4 Taking Confidence into Account

So far, we introduced the advantages of propagating imprecise information across layers and of considering the temporal dimension of situations. This aimed at addressing the challenges of a particular scenario for the *Cooking Assistant* application, which we described in §6.1.5.

Remember from §4.4 that a value of *confidence* is attached to every Context Attribute that is sent by the Perception layer to the Situation and Context Identification layer, providing an estimate of the uncertainty of the Perception layer output. This section evaluates the benefit of taking into account that confidence when reasoning about situations.

To this end, we first illustrate a case that challenges the recognition capabilities of the system. Then, we show how the propagation of the measure of confidence can help addressing such issue.

6.4.1 Need for Considering Confidence

There exist cases in which knowing that a Context Attribute has very low confidence can help improving the recognition of situations and preventing from providing unsuitable functionalities.

One of such cases is that in which all the Context Attributes have maximum confidence, except one, which has very low confidence (suggesting a wrong value). For example, consider again the challenging scenario presented in §6.1.5.

At some point in that scenario, the person in the kitchen seats quietly on a chair, resulting in the Sensing and Perception layer being unable to detect his presence. The Perception layer can decide to declare its ignorance by communicating the value $\{yes \bigcup no\}$ to the upper layer, which will handle that as we explained in §6.2.

However, it might be interesting to have an alternative to a sharp transition between a precise value (*e.g.*, $\{yes\}$) and an imprecise value ($\{yes \bigcup no\}$). This need is especially evident in this binary case, where the imprecise value corresponds to total ignorance.

6.4.2 The Flow of Confidence

The alternative is provided by the exchange of information about uncertainty: the Perception layer can in some cases hazard a guess about the presence of the person and associate a low confidence to that choice.

As previously illustrated, the level of confidence δ_i^t in a Context Attribute is provided by the Perception layer to specify the uncertainty of its output. Remember from Equation 4.4.1 that δ_i^t can be included in the Context Spaces reasoning, so as to reduce the overall confidence in the situation:

$$\mu(C^t, S, \sum, \Delta) = \sum_{i=1}^{n} w_i \kappa_i(a_i^t)\delta_i^t \tag{4.4.1}$$

Remember that the threshold ε, used as a decision-making criteria for triggering the Situation Space, was set to 1 (*cf.* §6.1.3). This value was chosen by considering the confidence in the situation occurrence when the Context Attributes have the desired values.

Now, the occurrence of the situation no longer depends on the Context Attribute values alone. The confidence on such values is also part of the reasoning algorithm. For this reason, ε must be modified.

Fixing the decision-making threshold

Consider the case in which the *Follow Cooking* Situation Space should be triggered. As we said, it corresponds to the combination of the following facts: nobody is in the kitchen, someone is in the living room and the hotplate is on. Let us consider that the Situation Space should be triggered even when the confidence in individual Context Attributes is not equal to one, although it should not be triggered if such confidence decreases too much.

Making empirical considerations that depend on the particular deployment, we can identify the worst case in which the Situation Space should still be triggered. For example, let us consider that this is represented by the following Context State:

$$C^t = \left\{ \begin{array}{ll} a_p^t = \{yes\} & \delta_p^t = 0.9 \\ a_k^t = \{no\} & \delta_k^t = 0.9 \\ a_h^t = \{ON\} & \delta_h^t = 0.9 \end{array} \right\}$$

In this case, the situation will have confidence:

$$\mu(C^t, S, \sum, \Delta) = \sum_{i=1}^{n} w_i \kappa_i(a_i^t)\delta_i^t = w_p \kappa_p(a_p^t)\delta_p^t + w_h \kappa_h(a_h^t)\delta_h^t + w_k \kappa_k(a_k^t)\delta_k^t$$

$$= 0.9w_p + 0.9w_h + 0.9w_k = 0.9$$

So, setting $\varepsilon = 0.9$ results in the fact that even Context Attributes with confidence lower than 1 can trigger the situation, although that confidence has to be reasonably high (*i.e.*, ≥ 0.9).

6.4.3 Benefits of the Confidence Flow

We now show the benefits of taking into account the impact of runtime confidence.

Consider again the challenging scenario illustrated in §6.1.5. When the person in the kitchen quietly seats on a chair, the Perception layer may communicate the following values and confidence for the interesting Context Attributes:

$$C^t = \left\{ \begin{array}{ll} a_p^t = \{yes\} & \delta_p^t = 1 \\ a_k^t = \{no\} & \delta_k^t = 0.6 \\ a_h^t = \{ON\} & \delta_h^t = 1 \end{array} \right\}$$

In this case, the Situation is not triggered, since its confidence is:

$$\mu(C^t, S, \sum, \Delta) = \sum_{i=1}^{n} w_i \kappa_i(a_i^t)\delta_i^t = 0.2\delta_p^t + 0.4\delta_h^t + 0.4\delta_k^t$$

$$= 0.2 + 0.4 + 0.4 \times 0.6 = 0.84 < \varepsilon = 0.9$$

So, although the values of the Context Attributes should trigger the situation, this does not happen because of the low confidence in one of them. This prevents from providing an unsuitable service, as the confidence in the emptiness of the kitchen is very low and suggests a recognition error.

6.4.4 Drawbacks of the Confidence Flow

As we showed, the exchange of information about the confidence in the Context Attributes' values allows avoiding the provision of unsuitable functionalities.

However, the introduction of such capability also raises some issues. These are mainly due to the lack of theoretical guarantees that the Situation Space will be triggered in and only in the right conditions. The reason behind this is the very high (virtually infinite) number of possible combinations of Context Attribute values and confidence.

The confidence on a Context Attribute, which depends on the sensor readings and on the models used in the sensor data fusion process, cannot be exactly mapped to the real-world conditions that determine its value. These are indeed extremely complex to identify and impossible to reproduce in most practical applications.

Recent work [BZ12] introduced an algorithm for the formal verification of Situation Space models. However, the uncertainty of Context Attribute values is not taken into account.

6.4.5 Finding a Compromise

To soften the problem, the confidence in the Context Attributes is bounded to a minimum value at the level of the Perception layer, due to the filtering algorithm (*cf.* §4.5.1).

More specifically, the confidence in the value of a Context Attribute never falls below a certain predefined threshold τ. Indeed, a value for the Context Attribute with higher confidence and lower precision is provided whenever the confidence on the considered value falls below the threshold. Even in the case of complete ignorance, the returned value has confidence equal to 1.

For instance, in the binary case of presence, when neither the confidence in the occupation nor the confidence in the emptiness of the kitchen are greater than τ, the Perception layer returns $a_k^t = \{yes \bigcup no\}$ and $\delta_k^t = 1$.

The threshold τ is fixed when deploying the system and is empirically chosen so as to guarantee that the Context Attribute values returned by the Perception layer are consistent with the real-world phenomenon. For instance, the expert deploying the system sets the threshold so that when somebody is in a room, the corresponding *Presence* Context Attribute returns the value $\{yes\}$ most of the time.

In our deployment, the threshold τ was fixed to 0.6. This means that the lowest possible confidence for any Context Attribute value is 0.6. This limits the fluctuations of the confidence and softens the issues caused by the lack of theoretical guarantees.

6.5 Scaling to Complex Real-World Applications

We described a simple context-aware application and showed how the improvements of the Context Spaces theory that we proposed can solve the challenges emerging when deploying the application on our Smart Home demonstrator.

We now present a quantitative assessment of the performance of the system. We aim at demonstrating that the system can scale when used to assist complex real-world situations and is deployed on real-world devices.

In particular, we focus on the performance of the Situation and Context Identification layer[1] in terms of time required for the computations on resource-constrained hardware. Such performance figures have to validate our choice of realizing a distributed reasoning architecture, which will be executed by future household appliances and devices.

[1]The performance figures of the Sensing and Perception layers of the system were assessed by Bastien Pietropaoli in the context of his doctoral research and showed promising results [PDW11].

This section describes the tests performed and presents the results. Then, we analyze these results and draw conclusions about the feasibility of adopting our system on large-scale applications.

6.5.1 Description of the Tests

The tests show the performance of ECSTRA, the adopted Context Spaces Theory implementation, when executed in different resource-constrained devices.

We measure the time required to reason about the occurrence of a Situation Space, when increasing the number of Context Attributes and the number of possible values per Context Attribute.

The number of Context Attributes is an interesting parameter that evaluates the performance degradation when the complexity of the Situation Spaces increases. Modeling more Context Attributes per Situation Space means more data to evaluate during the reasoning, therefore more complexity.

The number of possible values per Context Attribute expresses the complexity of each axis in the Situation Space model. In the tests, each axis (Context Attribute) is partitioned into intervals of values, each one bringing a different contribution to the decision of the situation (*cf.* §3.6.3). At runtime, the Context Attribute assumes a value that is contained in exactly one of these intervals, which will determine its contribution to the decision about the situation. For simplicity, in the remainder of this section, we will call possible *values* those that are actually possible *intervals*.

The reasoning is repeated on 10000 different Situation Spaces, obtained by random generation. For each Situation Space, the reasoning is repeated on 500 different Context States to obtain a measure of performance that is independent from the values of the Context Attributes. The Context State on which the reasoning is executed is randomly generated, by selecting a value per Context Attribute in the set of possible values.

Performance figures are calculated in terms of time necessary for the reasoning. The results are aggregated according to the number of Context Attributes and to the total number of possible values, obtained as the sum of the number of possible values of all the Context Attributes. For each test, we provide mean and standard deviation ($\sqrt{\frac{\Sigma(x-\bar{x})^2}{n-1}}$) of the reasoning time.

All tests were run on a laptop, on a Plug Computer and on a smartphone. In the remainder of this section, we first illustrate the results on the three devices and then analyze them.

6.5.2 Test Results on a Laptop

The computer is a Dell Latitude E4300 equipped with:

- Processor: Intel Core 2 Duo P9400 2.40 GHz;
- RAM: 4.00 GB;
- Operating system: Windows 7 Enterprise 64-bit;
- Execution environment: Java SE 1.6.0_21 with Java HotSpot 64-Bit Server VM (17.0-b17 built, mixed mode).

The test results are presented in Table 6.1 (all values are in milliseconds):

Num. val. / Num. CA	2-11	12-31	32-71	72-120
1-3	0.01 (± 0.004)	0.02 (± 0.005)	0.04 (± 0.009)	0.06 (± 0.010)
4-6	0.01 (± 0.004)	0.02 (± 0.005)	0.04 (± 0.008)	0.07 (± 0.010)
7-9	0.02 (± 0.003)	0.03 (± 0.006)	0.05 (± 0.009)	0.08 (± 0.010)
10-12	0.02 (± 0.004)	0.03 (± 0.007)	0.05 (± 0.009)	0.08 (± 0.010)
13-15	/	0.03 (± 0.006)	0.06 (± 0.009)	0.09 (± 0.011)

Table 6.1: ECSTRA performance test results on a laptop

6.5.3 Test Results on a Plug Computer

The Plug Computer is a DreamPlug, manufactured by Globalscale, with the following characteristics:

- Processor: 1.2GHz Marvell Kirkwood 88F6281;
- RAM: 512MB 16bit, DDR2-800 MHz;
- Operating system: Linux Ubuntu;
- Execution environment: OpenJDK Runtime Environment (IcedTea6 1.8.2) (6b18-1.8.2-4ubuntu1 9.04.1) + cacaovm (build 0.99.4, compiled mode).

The test results are presented in Table 6.2 (all values are in milliseconds).

Num. val. / Num. CA	2-11	12-31	32-71	72-120
1-3	0,7 (± 0,26)	1,8 (± 0,51)	4,3 (± 0,97)	7,9 (± 1,30)
4-6	0,7 (± 0,21)	1,9 (± 0,51)	4,3 (± 0,99)	7,9 (± 1,26)
7-9	0,7 (± 0,15)	1,8 (± 0,55)	4,4 (± 1,00)	8,0 (± 1,27)
10-12	0,7 (± 0,10)	1,8 (± 0,57)	4,5 (± 1,03)	8,2 (± 1,28)
13-15	/	1,8 (± 0,54)	4,4 (± 1,05)	8,2 (± 1,27)

Table 6.2: ECSTRA performance test results on a Plug Computer

6.5.4 Test Results on a Smartphone

The smartphone is an HTC Desire A8181, with the following characteristics:

- Processor: 1 GHz Scorpion, Adreno 200 GPU, Qualcomm QSD8250 Snapdragon chipset;
- RAM: 576 MB; ROM: 512 MB;
- Operating system: Android OS v2.2, kernel v. 2.6.32.15-gf5a401c.
- Execution environment: Dalvik VM

The test results are presented in Table 6.3 (all values are in milliseconds).

Num. CA \ Num. val.	2-11	12-31	32-50
1-3	0,3 (± 0,13)	0,8 (± 0,21)	1,5 (± 0,22)
4-6	0,4 (± 0,10)	0,9 (± 0,23)	1,6 (± 0,22)
7-10	0,5 (± 0,09)	1,0 (± 0,25)	1,7 (± 0,22)

Table 6.3: ECSTRA performance test results on a smartphone

6.5.5 Analysis of the Results of the Time-Performance Tests

The test results show the reasoning performance of ECSTRA on resource-constrained devices. The time required for reasoning on a situation never exceeds 10 milliseconds, for every device and situation that we considered.

The highly distributed reasoning architecture that we designed allows any device to reason about few situations only. Reasoning, indeed, is directly realized by a household appliance on the situations whose occurrence is likely to influence its behavior. Given these premises, the performance figures seem to guarantee the scalability of the reasoning architecture.

As it was predictable, the time needed for the reasoning process increases with the complexity of the Situation Space model. A more interesting result is that performance degrades much faster when increasing the number of possible values than when increasing the number of Context Attributes. So, designers should prefer Situation Space models with a large number of Context Attributes to others with few Context Attributes but many values per Context Attribute.

Such choice does not penalize the scalability of the overall architecture and is instead consistent with the requirements of the Perception layer. Having a limited number of possible Context Attribute values, indeed, also simplifies the task of that layer. The reason is that the complexity of sensor data fusion with the belief functions theory grows with the number of possible worlds [PDW11].

We also note that performance figures are worse on the Plug Computer than on the smartphone. Given the similar hardware characteristics, we attribute this to the different virtual machines by which ECSTRA code is run: cacaovm on the Plug Computer, Dalvik on the smartphone.

To summarize, the test results show that the reasoning architecture is responsive enough to be used for real-world applications, where appliances must quickly react to changes in context and situations. Such results back up the assumption that everyday appliances can reason about context and dynamically adapt their behavior.

6.6 Conclusions

This chapter validated the design and development choices of the Smart Home system that was presented by this doctoral dissertation.

To this end, we presented a context-aware application that was designed and developed using our interdisciplinary design approach. We showed how to build the computational models required by the application, focusing on the use of the Context Spaces Theory to reason about the occurrence of a real-world situation.

Then, we showed that the proposed context-aware system can face challenging scenarios, happening because of the limitations of any sensing technology. We explained how to address those issues, using the tools that were developed in the context of this doctoral research.

Namely, we described how uncertainty can be tolerated by propagating imprecise information, by introducing temporal aspects in the models of situations and by tuning such models to take into account the runtime assessment of confidence in the information that flows from lower to upper layers.

Finally, we demonstrated that our context-aware architecture can scale to complex real-world situations, while being executed by common resource-constrained devices in a distributed fashion.

The next chapter concludes the dissertation by summarizing the contribution of this doctoral research and by showing that the initial goal and objectives have been achieved.

Chapter 7

Conclusion

In the context of the societal challenges connected to the environmental questions, realizing energy savings is a topical issue. The domestic sector has been identified as one of the most promising ones, because of its high energy demand and its high potential for saving.

Some existing Smart Home initiatives have investigated the use of ICT and ubiquitous computing to realize energy saving at home. Unfortunately, the predominant approach is driven by technological considerations. In such an approach, the acceptability, appropriation and cost of the system for final users are rarely considered.

This doctoral work has aimed at demonstrating that a Smart Home can be designed with a human factor-driven approach, in order to maximize the chance of adoption of the system. For this, we collaborated with cognitive ergonomists to understand the point of view of users and to design functionalities that assist the activity while saving energy.

The collaboration allowed us to understand that domestic activity cannot be fully captured and understood by a computing system. The reason is that activity is built on engagements and concerns of home occupants, which cannot be observed. Thus, the system cannot make decisions on behalf of people.

7.1 Contributions

This doctoral work has designed and realized a context-aware system and the functionalities to be provided by it. These are conceived with the underlying assumption that the possible engagements and concerns of people are not directly observable. For this reason, the system does not decide on behalf of users.

7.1.1 First Objective — Interdisciplinary Design

To design and realize the functionalities to be provided by the Smart Home, we adopted an interdisciplinary approach.

For this, we collaborated with cognitive ergonomists for two years, in order to understand the point of view of users. We understood that domestic activity is the result of a continuous interaction between occupants, their cognitive processes and the environment, which become inseparable factors of the same reality. This convinced us that a computing system, external observer of such a complex phenomenon, cannot capture such complexity and automatically realize the proper actions.

For this, we concentrated on the potential of designing suitable interaction modalities, allowing users to easily select the preferred services, without disrupting their activity. We contributed to such process by exploiting the potential of augmented appliances to provide suitable services. We studied the characteristics of domestic appliances and of the interaction between them and their users. This allowed to design suitable functionalities that save energy and improve the comfort of home occupants.

The result of the interdisciplinary design is a description of future sustainable situations. These constitute a transformation of existing domestic activity, enabled by the Smart Home.

7.1.2 Second Objective — Realizing a Context-Aware System

To realize the sustainable situations, we designed and deployed a context-aware system. This uses sensing and reasoning capabilities to recognize real-world situations and provide suitable functionalities to home occupants.

To model and recognize real-world situations, we selected the simple and intuitive tools offered by the Context Spaces Theory. Such a theory provides powerful and expressive tools to model and recognize situations, while offering simple geometrical metaphors to illustrate the contextual dimensions and situations to non-technical audience. Such a choice eased the interdisciplinary collaboration and enabled the transition from the description of sustainable situations to the realization of the computing models required to realize them.

We contributed to the Context Spaces Theory with novel tools and approaches to deal with uncertain, imprecise and missing information. We also conceived and realized a set of tools to handle the temporal aspects of real-world situations. These measures were required to narrow the gap between the recognition capabilities of the system and the complexity of real-world domestic activity. More specifically, the newly developed tools allow designers to assess the risk of providing unsuitable functionalities and to instruct the system on the right strategies to react to uncertainty.

A paper presenting an overview of the novel approach to the management of uncertainty and imprecision was presented at the SAGAware workshop, co-located with UbiComp 2011 [DPW11]. The organizing committee invited us to submit an extended version of the paper, which was published in the International Journal of Pervasive Computing and Communications [DPW12].

Most of the improvements of the Context Spaces Theory were realized in collaboration with Andrey Boytsov, the designer and implementer of ECSTRA. The exchanges with Andrey started in 2010, when we met at ruSMART conference. Andrey joined our team at INRIA for an internship lasting two months.

The designed system was realized and deployed on the occasion of two demonstration sessions at EDF R&D, in Clamart (France). Different context-aware applications were shown, including that that was presented in this dissertation. The audience, composed of managers and researchers from different disciplines, gave very positive feedback.

7.1.3 Third Objective — Relying on Lightweight Instrumentation

In order to guarantee the acceptability of the Smart Home for final users, our context-aware system relies on lightweight instrumentation only. In particular, we exploit sensing, reasoning and actuating capabilities provided by augmented household appliances.

Based on the assumption that future houses will be equipped with communicating smart appliances, many software components of our Smart Home are executed by such appliances. They communicate information about their state, reason about contextual information and provide functionalities to users. Contextual information is also provided by environmental sensors. Several sensors are co-located in stand-alone communicating entities (the sensor nodes), which can be placed in key locations of the house. They take part in the sensing process and provide sensor readings through wireless communication.

Our contribution consists in having designed and developed the distributed applications and the computational models that provide adapted functionalities through augmented appliances. The performance figures on common devices have shown that such an approach is viable even for applications involving complex situations.

7.2 Perspectives

This doctoral research has investigated the design and realization of a Smart Home with an interdisciplinary approach. Future research directions can both be oriented towards improving our contributions and towards validating the approach in the real world. The remainder of this chapter enumerates the most viable possibilities.

7.2.1 Testing on Real Users

The final goal being that of deploying the system on real households, it would be interesting to test its acceptability for final users and the resulting energy saving. To this end, the system should be deployed on a testing platform accessible to potential users.

A collaboration with LOUSTIC[1] was established to open the way of such testing[2]. LOUSTIC is a multidisciplinary research platform that allows to observe the use of information and communication technologies by people. The platform includes a furnished apartment equipped with video and audio recording hardware that allows to observe the behavior of people when using the technology.

7.2.2 Improving the Management of Uncertainty

In the proposed context-aware system, the uncertainty of sensed information is assessed and communicated from layer to layer using a confidence value, attached to the information itself.

Currently, Context Attribute values are decided at the Perception layer and communicated to the Situation and Context Identification layer, together with a value of

[1] `http://www.loustic.net/`, accessed on 17 December 2012
[2] `http://www.loustic.net/infodiffuse`, accessed on 17 December 2012

confidence. A better alternative might consist in the Perception layer communicating the confidence in all the possible values of a Context Attribute, at a given moment.

In this way, no sharp decision would be made at the Perception layer, addressing the challenging scenario in which multiple values of the same Context Attribute have similar confidence. To support such improvement, the Context Spaces reasoning would have to be modified to take into account the confidence and contribution of several values of the same Context Attribute.

7.2.3 Recognizing Combinations of Situations

Currently, situations are recognized independently from each other. There is no way to model or recognize the fact that the occurrence of different situations is somehow connected, for example following a temporal development.

In domestic activity, situations can sometimes follow, precede or anyway happen with some temporal or causal connection with each other. This realizes a higher-order situation, made of combinations of basic situations, with (loose) temporal constraints. Our current system does not allow to model and recognize such higher-order situations.

To realize such functionality, we have started investigating some interesting tools offered by the Artificial Intelligence community. More specifically, existing research on Plan Recognition has attracted our attention.

We presented such ideas in a position paper, accepted at the GAPrec 2011 workshop [DFG+11], held in conjunction with the International Conference on Automated Planning and Scheduling (ICAPS).

The conference provided the opportunity to start collaborating with John Maraist, a senior researcher at SIFT LLC, Minneapolis, USA. Future work could explore the potential of combining Yappr [GMG08], a plan recognizer developed at SIFT, with the current capabilities of our context-aware system.

7.2.4 Improving the Communication Mechanism

In the implementation of the system, the adoption of Elvin as a Publish/Subscribe middleware and Avis as its implementation has shown some drawbacks. These are mainly due to the lack of guarantees on the delivery of notifications (*best effort* approach). In future developments, it might be interesting to guarantee the reliability of communications. A solution consists in adopting a Publish/Subscribe middleware that natively provides that feature. An example of such middleware is ØMQ[3].

[3] ØMQ — The Intelligent Transport Layer, http://www.zeromq.org/, Accessed 7 June 2012

Annexe A

Résumé de la thèse en Français

A.1 Introduction

En 2009, le secteur domestique à lui seul représentait le 30,9% de la consommation finale d'énergie en Europe [EEA12b]. Vues les questions environnementales d'aujourd'hui, ce travail de recherche vise à explorer les économies d'énergie qui peuvent être réalisées dans le secteur domestique.

Cette idée n'est pas nouvelle : depuis des années désormais, la *domotique* offre des solutions d'électronique qui visent à faciliter la vie domestique et, dans certains cas, à faire aussi des économies d'énergie.

Même des récents travaux de recherche ont exploré le potentiel de la technique afin de mettre en place ce qu'on appelle *maison intelligente* : une habitation équipée de technologies de l'information et de la communication (TIC) qui permettent de répondre aux besoins des occupants et de les anticiper [Ald03].

Malgré cet intérêt de l'industrie et de du monde académique, une large diffusion des maisons intelligentes est encore lointaine. La cause de cela a été identifiée en l'approche utilisée par les travaux existants, qui se sont souvent limités à explorer le potentiel technologique [Ald03] au lieu de s'intéresser aux personnes et à leurs besoins domestiques.

L'objectif de cette thèse a été de concevoir et réaliser une maison intelligente à partir de l'observation et compréhension de ses futurs occupants et de leur activité domestique. Pour ce faire, nos travaux ont pu bénéficier de la collaboration des psychologues ergonomes du département ICAME d'EDF R&D. Ceux-ci mènent des travaux de recherche visant à identifier comment les personnes prennent leurs décisions de consommation d'électricité et comment elles pourraient utiliser l'énergie de manière plus responsable [FG12].

A.1.1 Premier objectif : conception interdisciplinaire centrée utilisateur

Le **premier objectif** de cette thèse est de suivre une approche à la conception de type centré utilisateur, intégrant le point de vue sur l'activité humaine des experts en ergonomie cognitive.

Les défis qui découlent de cet objectif sont : s'approprier les modèles décrivant l'activité domestique, produits par les psychologues ergonomes, afin de comprendre le point de vue utilisateur. Notamment, un défi d'importance primordiale est la compréhension de le décalage qui existe entre le contexte utilisateur et le contexte machine. Autrement dit, le comportement humain est très ambigu et un système informatique qui l'observe depuis l'extérieur ne peut pas le comprendre dans toute sa complexité. Le défi devient donc de concevoir des fonctionnalités qui prennent en compte ce *décalage contextuel* (en anglais, *contextual gap*).

Le résultat attendu est la conception des futures situations domestiques comme des couplages entre les occupants et leur environnement, ce qui doit permettre de réaliser des économies d'énergie, tout en préservant le confort de vie.

A.1.2 Deuxième objectif : conception et réalisation d'un système sensible au contexte

Le **deuxième objectif** de la thèse est de concevoir et de réaliser un système informatique qui fournit des fonctionnalités permettant de transformer les situations existantes en *situations durables*. Ceci se traduit en la définition et réalisation d'une **architecture matérielle et logicielle capable de reconnaitre les situations et de fournir des fonctionnalités qui les transforment**.

Les défis relatifs à cet objectif sont en lien avec l'approche interdisciplinaire adoptée. Plus précisément, les fonctionnalités conçues doivent être transposées vers une spécification technique qui prenne en compte le décalage entre contexte utilisateur et contexte machine. Afin de vérifier cette propriété, le système devra utiliser des outils techniques s'appuyant sur des abstractions intuitives. Ceci devra permettre aux psychologues ergonomes de vérifier que le système est conçu de la manière attendue. Afin de prendre en compte le décalage contextuel, le future système devra aussi faire face à de l'information incertaine, imprécise et manquante.

A.1.3 Troisième objectif : intégrer l'acceptabilité aux critères de conception du système

Le **troisième objectif** de cette thèse est de prendre en compte l'acceptabilité du futur système comme un critère fondamental. Ceci se reflète notamment sur le choix de l'instrumentation, qui évitera les technologies de capture intrusives, telles que les caméras ou les microphones.

La reconnaissance de contexte et de l'activité des habitants devient plus complexe avec une instrumentation limitée. Toutefois, un nouveau type d'instrumentation se rend de plus en plus disponible dans les maisons d'aujourd'hui : les électroménagers augmentés, tels que les fers à repasser qui détectent leurs mouvements ou les téléviseurs qui détectent la présence des personnes pour se mettre en veille quand ils ne sont pas utilisés. Le défi devient donc d'**exploiter les électroménagers pour réaliser un réseau de dispositifs sensibles au contexte qui échangent de l'information contextuelle et qui utilisent leurs fonctionnalités de manière coordonnée pour fournir des services adaptés aux habitants**.

A.1.4 Structure de ce résumé

Le reste de ce résumé est organisé en quatre sections principales : nous décrivons d'abord une classification des maisons intelligentes existantes et illustrons leurs limitations (*cf.* §A.2). Nous présentons ensuite une nouvelle approche interdisciplinaire centrée utilisateur, développée dans le cadre de ce travail de thèse (*cf.* §A.3). Ensuite, nous décrivons l'architecture de capture de contexte et de fourniture de services que nous avons conçue et réalisée (*cf.* §A.4). Pour finir, nous présentons une des applications qui ont permis de valider notre démarche (*cf.* §A.5). Nous concluons ce résumé avec des conclusions et perspectives futures (*cf.* §A.6).

A.2 Maisons Intelligentes : Uniquement de la Technique ?

En cette section, nous montrons que l'acceptabilité de nombreuses maisons intelligentes est compromise par l'absence d'une méthodologie de conception dirigée par des considérations ergonomiques et s'appuyant sur une analyse rigoureuse de l'activité domestique. Nous montrons ensuite une méthodologie qui répond aux limitations des approches existantes.

Les travaux existants dans le domaine des maisons intelligentes peuvent être classifiés selon trois catégories [Jen] : les maisons *contrôlables*, les maisons *programmables* et les maisons *intelligentes*. Pour chacune d'entre elles, nous présenterons leurs caractéristiques par rapport à leur instrumentation, à leur façon de traiter le contexte et aux services rendus.

A.2.1 Maisons contrôlables

Les maisons contrôlables permettent aux habitants de contrôler différents dispositifs de manière avancée : avec une télécommande centralisée, avec des interfaces avancées (*e.g.*, reconnaissance de parole ou de geste) ou en connectant différents dispositifs entre eux pour qu'ils échangent de l'information (*e.g.*, reproduction multimédia à partir d'un système de stockage distant).

Les maisons contrôlables sont principalement le résultat de l'industrie domotique, même si certains travaux de recherche ont aussi visé les mêmes problématiques (*e.g.*, [Oxy]). Elles permettent d'améliorer la façon dont les habitants peuvent contrôler les équipements. Toutefois, leur conception est dirigée par des considérations technologiques, le but étant celui de regrouper les dispositifs présents dans une maison.

Par conséquence, même si les maisons contrôlables peuvent fournir des fonctionnalités intéressantes, elles ne sont pas conçues pour s'intégrer souplement dans l'activité domestique.

A.2.2 Maisons programmables

Les maisons programmables permettent de programmer des dispositifs pour qu'ils s'allument, s'éteignent ou exécutent des opérations prédéterminées lorsque certaines conditions se produisent. Les maisons programmables peuvent réagir à des événements déclenchés par des simples capteurs ou alors reconnaitre des combinaisons complexes d'événements qui correspondent à des situations. Par exemple, une lumière peut être allumée au passage d'une personne ou alors une chaîne radio peut s'éteindre lorsqu'une personne utilise un ordinateur dans la même pièce, afin de faciliter la concentration. Exemples de maisons programmables sont celle issue du projet européen *SM4All* (Smart Homes for All) [AAB+11, SM4] et la *Gator Tech Smart House* [HMEZ+05].

Les maisons programmables s'appuient sur une correspondance rigide entre conditions et actions du système. L'analyse de l'activité domestique, au contraire, montre que les situations de vie sont caractérisées par une forte variabilité. Par conséquent, les maisons programmables fournissent souvent des fonctionnalités peu ou pas adaptées à l'activité domestique.

A.2.3 Maisons intelligentes

Les maisons intelligentes changent leur comportement automatiquement, en fonction des habitudes des habitants. Leur fonctionnement peut être reconduit à l'observation des pratiques quotidiennes et à l'émulation des choix des personnes en fonction des situations détectées. Exemples de maisons intelligentes sont celles issues des projets ERGDOM [CDRE+03, Bon08] et Adaptive House [Moz04, Moz98].

Les maisons intelligentes cherchent à détecter l'intention et l'« etat intérieur » des habitants en s'appuyant sur les observations passées, afin de reproduire des comportement observés dans les situations analogues. Toutefois, l'analyse de l'activité montre qu'il n'est pas possible de détecter de manière fiable le contexte utilisateur à partir d'une observation externe. Par exemple, si le système a observé qu'un habitant éteint la radio quand il utilise l'ordinateur, ceci ne peut pas être reconduit à un automatisme à réaliser à chaque fois, car les conditions qui amènent la personne à choisir peuvent varier.

Cette observation soulève aussi le problème des automatismes : vue l'observation que l'on vient de faire, un système où toutes les décisions sont prises de manière automatique est voué à l'échec. Ceci est d'autant plus vrai quand ces automatismes s'appuient sur l'identification de régularités dans l'activité domestique. Le problème d'une telle approche, en effet, est que les habitudes domestiques ne sont pas des reproductions identiques de séquences d'actions prédéterminées mais plutôt des préoccupations récurrentes des habitants, qui se manifestent à chaque fois de manière différente [FG12].

A.3 La Thèse : une Approche Interdisciplinaire Centrée Utilisateur

Cette thèse vise à concevoir des maisons intelligentes avec une approche centrée utilisateur, afin de dépasser les limites des maisons intelligentes existantes, que nous venons de présenter. En particulier, **cette thèse vise à démontrer qu'il est possible de concevoir et réaliser un système qui, même si incapable de capturer et comprendre le point de vue utilisateur, produit des fonctionnalités qui le respectent**.

A.3.1 Apports de l'ergonomie cognitive

Pour ce faire, nous sommes partis de considérations à propos des besoins des habitants et de leur activité. Cette tâche a été assistée par la collaboration avec des ergonomes cognitifs. L'*ergonomie* est une discipline scientifique qui vise à comprendre les interactions entre les humains et d'autres éléments d'un système [ERG]. Plus spécifiquement, l'ergonomie *cognitive* étudie les façons dont ces interactions sont influencées par les processus mentaux, tels que la perception, la mémoire, le raisonnement et la réponse motrice.

Des études ergonomiques de l'activité domestique en situation [FG12, GVRG+11, DFG+11, ZF10, PFH09, SDFH09, DSFH05] ont fourni la base et la boussole pour orienter cette thèse en informatique.

A.3.2 Apports de l'Informatique Diffuse

Afin d'assister l'activité domestique sans la géner, il est évident que la maison intelligente doit agir de manière *invisible*, en réduisant au minimum les interactions explicites. Ceci est aussi l'objectif de l'*Informatique Diffuse* (en anglais, *Ubiquitous Computing*) [Wei93].

Suivant ce paradigme d'interaction homme-machine, de nombreux calculateurs s'integrent à l'environnement physique et fournissent aux personnes des fonctionalités adaptées au contexte. Ceci est réalisé de façon transparente, en exigeant peu ou pas d'interactions explicites avec les utilisateurs.

A.3.3 Conception de *situations durables*

Grâce aux apports de l'informatique diffuse et de l'ergonomie cognitive, il est possible de rapprocher les situations de vie domestiques aux exigeances du développement durable. Pour ce faire, nous pouvons concevoir de nouvelles situations qui facilitent l'activité domestique et, en même temps, les comportements éco-responsables, en particulier vis-á-vis des économies d'énergie. Nous appelons cela des *situations durables*.

Afin de concevoir des situations durables, une nouvelle méthodologie de conception interdisciplinaire a été développée en collaboration avec des psychologues ergonomes. Cette méthodologie inclue trois phases :

1. L'**analyse de l'activité domestique en situation**, réalisée par des psychologues ergonomes, permet de gagner une connaissance fine de l'activité. Cette analyse montre que l'activité domestique est opportuniste et constituée en même temps de différents préoccupations et engagements, qui déterminent le comportement et rendent son explication très complexe. L'activité ne suit jamais un plan hierarchique et prétabli ; même les routines ne sont que des préoccupations recurrentes. Les échelles temporelles, spatiales et individuelle/collective s'entrelacent.

2. La **définition des fonctionnalités possibles** permet d'identifier les pratiques à préserver et celles qui doivent évoluer : les préoccupations recurrentes des habitants peuvent être reconnues, afin de réaliser des automatismes ; les variations contextuelles du comportement peuvent être reconnues et gérées différemment (*e.g.*, en donnant la main à l'utilisateur) ; les situations futures ainsi que les

points de vue individuels/collectifs et locaux/globaux peuvent être pris en compte afin de concevoir des fonctionnalitées plus adaptées. Les **charactéristiques des électroménagers et des autres appareils présents dans un foyer** peuvent aussi être pris en compte : elles fournissent des indications sur le comportement d'une personne qui utilise un appareil et, en même temps, des critères pour concevoir des fonctionnalités adaptées lorsque le comportement attendu ne se présente pas. En complément à cela, la **définition du potentiel technique** permet d'encrer la conception des fonctionnalités à la faisabilité technique. Cela passe par la définition de l'architecture d'un système informatique qui respecte des contraintes d'acceptabilité, tout en étant en mesure de reconnaître des situations domestiques.

3. La **transformation des situations actuelles en situations durables** se réalise en combinant la connaissance des pratiques existantes et des fonctionnalités visées (tâche de l'ergonome cognitif) avec l'estimation du potentiel technique (tâche de l'informaticien). Le résultat est un scénario décrivant les futures situations durables, du point de vue du système. Cela permet de passer facilement de cette dernière phase de conception à la réalisation technique. Les modalités d'interaction homme-machine sont aussi spécifiées, afin de gérer les situations où le système est incapable de prendre des décisions automatiques, suite à l'impossibilité de comprendre la complexité de l'activité humaine (ce que nous avons appelé décalage contextuel ou *contextual gap*).

A.4 Contribution Technique : une Architecture Sensible au Contexte

Nous avons jusqu'ici illustré comment concevoir des situations durables, démontrant ainsi que les fonctionnalités offertes par une maison intelligente peuvent prendre en compte le point de vue de l'utilisateur. Nous analysons maintenant les caractéristiques techniques d'un système qui fournit ces fonctionnalités.

Pour cela, nous devons introduire une capacité indispensable à la réalisation des fonctionnalités : la sensibilité au contexte (en anglais, *context awareness*). Un système est sensible au contexte s'il ≪ utilise le contexte pour fournir des informations et/ou des services pertinents à l'utilisateur, où la pertinence dépend de la tâche de l'utilisateur ≫ [Dey01].

A.4.1 Techniques de reconnaissance de contexte et de situations

Pour réaliser un système sensible au contexte, nous avons d'abord étudié les techniques existantes de *reconnaissance de contexte et de situations*. Celles-ci représentent l'évolution des approches initiales, où les données brutes des capteurs étaient utilisées directement pour développer les applications. Nous assistons désormais à la diffusion d'approches s'appuyant sur des modèles de contexte, où le traitement des données brutes est réalisé par des techniques de *fusion de données* et les développeurs d'applications peuvent s'appuyer sur des abstractions de haut niveau, correspondantes au contexte et aux situations.

Parmi les techniques qui constituent l'état de l'art, nous avons retenu la théorie des *Context Spaces*[Pad06]. Il s'agit d'une technique qui permet de *spécifier* (par opposition aux techniques s'appuyant sur l'*apprentissage artificiel*) les situations à reconnaître et qui utilise des simples abstractions géométriques pour décrire le contexte et les situations. La théorie fournit des outils de modélisation intuitifs, qui facilitent la collaboration interdisciplinaire, en permettant aux chercheurs aux différents compétences de discuter autour de concepts simples. La Théorie des Context Spaces permet la spécification d'un large spectre de caractéristiques des situations, avec une structure hiérarchique. Elle fournit également des algorithmes de traitement de l'information qui gèrent l'incertitude et les aspects flous du contexte, ainsi que le manque d'information. Enfin, elle permet l'utilisation de techniques supplémentaires, y compris des techniques pour la validation formelle de modèles [BZ12] et pour la prédiction de contexte [BZS09].

A.4.2 Réalisation d'une architecture sensible au contexte

Nous avons réalisé un système sensible au contexte qui s'appuie sur la théorie des Context Spaces pour reconnaître les situations. Pour ce faire, nous avons défini une architecture de capture/traitement d'information contextuelle et de fourniture de fonctionnalités adaptées.

La conception de l'architecture a été guidée par des principes d'acceptabilité : l'instrumentation est limitée et n'inclue pas de technologies comme les capteurs portables, les caméras ou les microphones. A leur place, nous avons bâti l'architecture sur des capteurs disséminés dans l'environnement et sur des électroménagers *augmentés*.

L'abstraction de l'information contextuelle a été réalisée en couches successives, afin d'utiliser des techniques adaptées à chaque type d'information à traiter [CCDG05]. Plus précisement, la couche appelée de capture ou *sensing*, constituée de capteurs et électroménagers augmentés, fournit des données brutes qui sont traitées par la couche appelée *perception*. Cette dernière s'appuie sur la *théorie des fonctions de croyance* ou *Dempster-Shafer* [Sha76] pour produire des abstractions qui sont fournie à la théorie des Context Spaces, qui constitue la couche dite d'*identification du contexte et des situations*. La dernière couche, d'*exploitation*, distribue l'information produite par les Context Spaces aux électroménagers, appareils et calculateurs qui fournissent les fonctionnalités aux utilisateurs.

La théorie des Context Spaces a été modifiée et améliorée pour s'integrer proprement à l'architecture présentée. En particulier, la capacité à s'appuyer sur les abstractions produites par la couche de perception, qui réalise la fusion des données brutes, a été conçue et réalisée. Vue l'incertitude qui caractérise tout système de capture et traitement de l'information contextuelle, des mécanismes de prise en compte et propagation de mesures de confiance, imprécision et ignorance ont été mises au point ou améliorés. Des nouveaux outils de gestion de la dimension temporelle des situations ont été integré à la théorie et réalisés.

A.5 Validation de la Contribution

Le système a été réalisé en collaboration avec d'autres membres de l'équipe de recherche au sein de laquelle ce travail de thèse a été mené. Un local à deux pièces a été instrumenté et utilisé comme plateforme d'expérimentation. Nous avons utilisé des ordinateurs embarqués, des nœuds de capteurs sans fils et des appareils tels que des smartphones et des tablettes numériques. Les capteurs mesurent des propriétés telles que le mouvement, le niveau sonore et les vibrations. La reconnaissance des situations et la fourniture des fonctionnalités aux utilisateurs sont réalisées directement par les appareils concernés, en exploitant la Théorie des Context Spaces. Une implémentation existante des Context Spaces (*ECSTRA*) a été modifiée, améliorée et integrée au système.

La contribution de cette thèse a été validée grâce à la réalisation et au test de plusieurs applications sensibles au contexte. Par exemple, nous avons réalisée une application qui assiste la cuisson d'un repas. Ceci est rendu possible par l'exploitation d'une plaque augmentée, qui communique son état aux autres appareils du système, ainsi que par un écran tactile, localisé dans une autre pièce, qui permet d'arrêter la plaque à distance lorsque le système détecte une absence prolongée de la cuisine.

Les applications ont été conçues avec la méthodologie interdisciplinaire présentée et elles ont été réalisées en exploitaint l'architecture informatique décrite. La Théorie des Context Spaces a été utilisée pour reconnaître les situations qui déclanchent les fonctionnalités (*e.g.*, proposition d'éteindre la plaque). Les situations de mise en échec du système, dues principalement à l'incertitude intrinsèque à tout mécanisme de capture et reconnaissance de contexte, ont été gérées grâces aux nouveaux outils conçus et réalisés par ce travail de thèse.

Les tests réalisés ont aussi montré que le système peut facilement être appliqué à des cas réels, où les situations et fonctionnalités à gérer peuvent être très nombreuses. En particulier, les choix réalisés permettent de fournir des fonctionnalités complexes en s'appuyant sur des appareils aussi répandus que les tablettes et les smartphones.

A.6 Conclusions

Dans le contexte des défis sociétaux liés aux questions environnementales, réaliser des économies d'énergie est une question d'actualité. Le secteur domestique a été identifié comme l'un des plus prometteurs, en raison de sa forte demande d'énergie et son potentiel d'économies.

Certaines initiatives de Smart Home existantes ont enquêté sur l'utilisation de l'informatique diffuse pour réaliser des économies d'énergie à la maison. Malheureusement, la pluspart des approches part de considérations technologiques, en prenant rarement en compte l'acceptabilité, l'appropriation et le coût du système pour les utilisateurs finaux.

Ce travail de thèse a cherché à démontrer qu'une maison intelligente peut être conçue avec une approche axée sur le facteur humain, de façon à maximiser la probabilité d'adoption du système. Pour cela, nous avons collaboré avec des ergonomes cognitifs afin de comprendre le point de vue des utilisateurs et de concevoir des fonctionnalités qui facilitent l'activité, tout en économisant de l'énergie.

A.6.1 Apports

Un système sensible au contexte et ses fonctionnalités ont été conçus et réalisés avec l'hypothèse sous-jacente que les engagements et les préoccupations des personnes ne sont pas directement observables. Ceci a aussi abouti à la définition d'une méthodologie pour la conception de telles fonctionnalités, en collaboration avec des psychologues ergonomes d'EDF R&D.

Plus spécifiquement, **nous nous sommes concentrés sur le potentiel de concevoir des modaltés d'interaction appropriées, ce qui permet aux utilisateurs de sélectionner facilement les services préférés, sans perturber leur activité**. Nous avons contribué à ce processus en exploitant le potentiel des appareils augmentés pour fournir des services appropriés. Nous avons étudié les caractéristiques des électroménagers et de l'interaction entre ceux-ci et leurs utilisateurs. Cela a permis de concevoir fonctionnalités adaptées qui permettent d'économiser de l'énergie et d'améliorer le confort des occupants de la maison.

Nous avons contribué à la Théorie des Context Spaces en l'appliquant à un niveau d'abstraction différent et en concevant des nouveaux outils qui permettent de traiter des informations incertaines, imprécises et manquantes, ainsi que les outils pour gérer les aspects temporels des situations du monde réel. Le nouveaux outils permettent aux concepteurs d'évaluer le risque de fournir des fonctionnalités inadaptées et d'apprendre au système les bonnes stratégies pour réagir à l'incertitude.

Nous avons conçu et développé des applications et des modèles informatiques qui fournissent des fonctionnalités adaptées en exploitant des électroménagers augmentés, ce qui facilite l'acceptabilté et la diffusion du système proposé. Les performances sur des dispositifs communs ont montré qu'une telle approche est viable, même pour des applications impliquant des situations complexes.

A.6.2 Collaborations et dissémination

Parmi les communications réalisées dans le cadre de ce travail, un document présentant un aperçu de la nouvelle approche de la gestion de l'incertitude et de l'imprécision a été présenté lors du workshop SAGAware, co-localisés avec UbiComp 2011 [DPW11]. Le comité organisateur nous a ensuite invités à soumettre une version étendue du document, qui a été publié dans l'*International Journal of Pervasive Computing et Communications* [DPW12].

La plupart des améliorations de la Théorie des Context Spaces ont été réalisés en collaboration avec Andrey Boytsov, le concepteur et développeur de ECSTRA. Les échanges avec Andrey ont commencé en 2010, lorsque nous nous sommes rencontré à la conférence ruSMART. Andrey a rejoint notre équipe à l'INRIA pour un stage d'une durée de deux mois.

Le système conçu a été déployé à l'occasion de deux démonstrations à EDF R&D, à Clamart. Différentes applications sensibles au contexte ont été montrées, y comprise celle qui a été présenté dans ce document. Le public, composé de dirigeants et chercheurs en différentes disciplines, a donné un retour très positif.

A.6.3 Perspectives

Cette thèse de doctorat a étudié la conception et la réalisation d'une maison intelligente avec une approche interdisciplinaire. **Les axes de recherche future peuvent être orientés à la validation de la démarche avec des vrais utilisateurs/habitants**.

Il serait par exemple intéressant de tester son acceptabilité pour les utilisateurs finaux et les économies d'énergie réalisées. A ce fin, le système devrait être déployé sur une plate-forme de test accessible aux potentiels utilisateurs. Une collaboration (déjà en place) avec LOUSTIC, une plateforme de recherche multi-disciplinaire qui permet d'observer l'utilisation des TIC par les personnes, pourrait permettre de réaliser de tels tests[1].

Des futures recherches peuvent aussi viser l'amélioration des contributions réalisées. En particulier, il pourrait être intéressant de réaliser des mécanismes de reconnaissance de situations composées, par exemple lorsque les situations s'entrelacent temporellement. Pour réaliser une telle fonctionnalité, nous regardons avec intérêt la recherche existante sur la Reconnaissance de Plan. Nous avons présenté des idées préliminaires dans une communication acceptée à GAPrec 2011 [DFG$^+$11], un workshop organisé dans le cadre de ICAPS. La conférence a fourni l'occasion de commencer à collaborer avec John Maraist, *senior researcher* à SIFT LLC, Minneapolis, Etats-Unis. Les travaux futurs pourrait explorer le potentiel de combiner Yappr [GMG08], un algorithme de reconnaissance de plan mis au point chez SIFT, avec les capacités actuelles de notre système sensible au contexte.

[1] `http://www.loustic.net/infodiffuse`

Bibliography

[AAB+11] M. Aiello, F. Aloise, R. Baldoni, F. Cincotti, C. Guger, A. Lazovik, M. Mecella, P. Pucci, J. Rinsma, G. Santucci, and M. Taglieri. Smart homes to improve the quality of life for all. In *Engineering in Medicine and Biology Society,EMBC, 2011 Annual International Conference of the IEEE*, pages 1777 –1780, 30 2011-sept. 3 2011.

[AJLS97] M.D. Addlesee, A. Jones, F. Livesey, and F. Samaria. The ORL active floor. *Personal Communications, IEEE*, 4(5):35 –41, oct 1997.

[Ald03] Frances Aldrich. Smart homes: Past, present and future. In Richard Harper, editor, *Inside the Smart Home*, pages 17–39. Springer London, 2003.

[ALM+05] Juan C. Augusto, Jun Liu, Paul Mccullagh, Hui Wang, and Jian bo Yang. Management of uncertainty and spatio-temporal aspects for monitoring and diagnosis in a smart home. Technical report, 2005.

[ANH07] C.B. Anagnostopoulos, Y. Ntarladimas, and S. Hadjiefthymiades. Situational computing: An innovative architecture with imprecise reasoning. *Journal of Systems and Software*, 80(12):1993 – 2014, 2007.

[AS97] Varol Akman and Mehmet Surav. The use of situation theory in context modeling. *Computational Intelligence*, 13:427–438, 1997.

[BB08] Lynne Baillie and David Benyon. Place and technology in the home. *Computer Supported Cooperative Work (CSCW)*, 17:227–256, 2008. 10.1007/s10606-007-9063-2.

[BBH+10] Claudio Bettini, Oliver Brdiczka, Karen Henricksen, Jadwiga Indulska, Daniela Nicklas, Anand Ranganathan, and Daniele Riboni. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing*, 6(2):161 – 180, 2010.

[BCE+07] S. Bonhomme, E. Campo, D. Esteve, J. Guennec, and T. LAAS-CNRS. An extended prosafe platform for elderly monitoring at home. pages 4056–4059, 2007.

[BCPB04] Michel Banâtre, Paul Couderc, Julien Pauty, and Mathieu Becus. Ubibus: Ubiquitous computing to help blind people in public transport. 3160:535–537, 2004.

[Bon08] S. Bonhomme. *Méthodologie et outils pour la conception d'un habitat intelligent*. PhD thesis, Institut National Polytechnique de Toulouse, 2008.

[BZ10] Andrey Boytsov and Arkady Zaslavsky. Extending context spaces theory by proactive adaptation. 6294:1–12, 2010.

[BZ11a] Andrey Boytsov and Arkady Zaslavsky. From sensory data to situation awareness: Enhanced context spaces theory approach. *Dependable, Autonomic and Secure Computing, IEEE International Symposium on*, 0:207–214, 2011.

[BZ11b] Andrey Boytsov and Arkady B. Zaslavsky. ECSTRA - distributed context reasoning framework for pervasive computing systems. pages 1–13, 2011.

[BZ12] Andrey Boytsov and Arkady Zaslavsky. Formal verification of context and situation models in pervasive computing. *Pervasive and Mobile Computing*, (0):–, 2012. doi:10.1016/j.pmcj.2012.03.001.

[BZS09] Andrey Boytsov, Arkady Zaslavsky, and Kåre Synnes. Extending context spaces theory by predicting run-time context. In *NEW2AN '09 and ruSMART '09: Proceedings of the 9th International Conference on Smart Spaces and Next Generation Wired/Wireless Networking and Second Conference on Smart Spaces*, pages 8–21, Berlin, Heidelberg, 2009. Springer-Verlag.

[CB03] P. Couderc and M. Banatre. Ambient computing applications: an experience with the SPREAD approach. In *Proc. 36th Annual Hawaii International Conference on System Sciences*, page 9pp., 6–9 Jan 2003.

[CC02] E. Campo and M. Chan. Detecting abnormal behaviour by real-time monitoring of patients. pages 8–12, 2002.

[CCDG05] Joëlle Coutaz, James L. Crowley, Simon Dobson, and David Garlan. Context is key. *Commun. ACM*, 48:49–53, March 2005.

[CCE03] É. Campo, M. Chan, and D. Estève. L'apprentissage des modes de vie: une base indispensable au développement d'un habitat "intelligent". *Annals of Telecommunications*, 58(5):850–865, 2003.

[CDRE+03] E. Campo, J.P.S. Di Rinaldi, D. Estève, N. Bailly, and F. Benard. Développement d'une nouvelle génération de gestionnaire d'énergie autoconfigurable pour l'habitat: le concept ergdom. In *Annales du Bâtiment et des Travaux Publics*, pages 1270–9840, 2003.

[CHRC95] M. Chan, C. Hariton, P. Ringeard, and E. Campo. Smart house automation system for the elderly and the disabled. In *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on*, volume 2, pages 1586–1589 vol.2, Oct 1995.

[CN09] Liming Chen and Chris D. Nugent. Ontology-based activity recognition in intelligent pervasive environments. *IJWIS*, 5(4):410–430, 2009.

[CR04] Andy Crabtree and Tom Rodden. Domestic routines and design for the home. *Computer Supported Cooperative Work (CSCW)*, 13:191–220, 2004. 10.1023/B:COSU.0000045712.26840.a4.

[CWkYZf05] Liang-zhou Chen, Shi Wen-kang, Deng Yong, and Zhu Zhen-fu. A new fusion approach based on distance of evidences. *Journal of Zhejiang University - Science A*, 6:476–482, 2005. 10.1007/BF02839418.

[CYD06] Diane Cook, Michael Youngblood, and Sajal Das. A multi-agent approach to controlling a smart environment. 4008:165–182, 2006.

[CYH+03] D.J. Cook, M. Youngblood, III Heierman, E.O., K. Gopalratnam, S. Rao, A. Litvin, and F. Khawaja. Mavhome: an agent-based smart home. pages 521 – 524, march 2003.

[DCB+02] S.K. Das, D.J. Cook, A. Battacharya, III Heierman, E.O., and Tze-Yun Lin. The role of prediction algorithms in the mavhome smart home architecture. *Wireless Communications, IEEE*, 9(6):77 – 84, dec. 2002.

[Dey01] Anind K. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5:4–7, 2001. 10.1007/s007790170019.

[DFG+11] M. Dominici, M. Fréjus, J. Guibourdenche, B. Pietropaoli, and F. Weis. Towards a system architecture for recognizing domestic activity by leveraging a naturalistic human activity model. In *Workshop on Goal, Activity and Plan Recognition at the International Conference on Automated Planning and Scheduling (ICAPS)*, 2011.

[DHKZG08] Pari Delir Haghighi, Shonali Krishnaswamy, Arkady Zaslavsky, and Mohamed Medhat Gaber. Reasoning about context in uncertain pervasive computing environments. pages 112–125, 2008.

[DLN] http://www.dlna.org/. Accessed on 18 December 2012.

[DP88] Didler Dubois and Henri Prade. Representation and combination of uncertainty with belief functions and possibility measures. *Computational Intelligence*, 4(3):244–264, 1988.

[DP01] D. Dubois and H. Prade. La problématique scientifique du traitement de l'information. *Information-Interaction-Intelligence*, 1(2), 2001.

[DPW11] M. Dominici, B. Pietropaoli, and F. Weis. Towards a feasibility-driven uncertainty-aware layered architecture for recognizing complex domestic activity. In *Proceedings of the 2011 International Workshop on Situation Activity & Goal Awareness (SAGAware) at the 13th ACM International Conference on Ubiquitous Computing (UbiComp)*, pages 89–94. ACM, 2011.

[DPW12] Michele Dominici, Bastien Pietropaoli, and Frédéric Weis. Experiences in managing uncertainty and ignorance in a lightly instrumented smart home. *International Journal of Pervasive Computing and Communications*, 8(3):225–249, 2012.

[DSFH05] J. Dugdale, P. Salembier, M. Frejus, and Y. Haradji. An analysis of empirical studies of domestic activities. descriptive framework and context modeling. Technical report, EDF-IRIT, Clamart, France, 2005.

[DZWB10] Michele Dominici, Giulio Zecca, Frédéric Weis, and Michel Banâtre. Physical approach in smart homes: a proposition and a prototype. In *Proceedings of the Third International Conference on Smart Spaces, ruSMART 2010, and 10th International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking, NEW2AN 2010*, ruSMART/NEW2AN'10, pages 111–122, Berlin, Heidelberg, 2010. Springer-Verlag.

[EB09] Michael Eckert and François Bry. Complex event processing (CEP). (PMS-FB-2009-6), 2009.

[EEA12a] Final energy consumption by sector (csi 027/ener 016) - assessment published mar 2012. Technical report, European Environment Agency (EEA), March 2012.

[EEA12b] Total final energy consumption by sector in the eu-27, 1990-2009. Technical report, European Environment Agency (EEA), March 2012.

[EG01] W. Keith Edwards and Rebecca E. Grinter. At home with ubiquitous computing: Seven challenges. In *Proceedings of the 3rd international conference on Ubiquitous Computing*, UbiComp '01, pages 256–272, London, UK, UK, 2001. Springer-Verlag.

[ELV] Elvin protocol suite. http://elvin.org/specs/index.html. Accessed on 25 September 2011.

[ERG] What is ergonomics. `http://iea.cc/01_what/What is Ergonomics.html`. Accessed on 25 June 2012.

[FG12] M. Fréjus and J. Guibourdenche. Analysing domestic activity to reduce household energy consumption. *Work: A Journal of Prevention, Assessment and Rehabilitation*, 41:539–548, 2012.

[GCTL10] Tao Gu, Shaxun Chen, Xianping Tao, and Jian Lu. An unsupervised approach to activity recognition and segmentation based on object-use fingerprints. *Data & Knowledge Engineering*, 69(6):533 – 544, 2010.

[GGH06] Björn Gottfried, Hans Guesgen, and Sebastian Hübner. Spatiotemporal reasoning for smart homes. In Juan Augusto and Chris Nugent, editors, *Designing Smart Homes*, volume 4008 of *Lecture Notes in Computer Science*, pages 16–34. Springer Berlin / Heidelberg, 2006.

[GMG08] Christopher W. Geib, John Maraist, and Robert P. Goldman. A new probabilistic plan recognition algorithm based on string rewriting. In *ICAPS*, pages 91–98, 2008.

[GPZ05] Tao Gu, Hung Keng Pung, and Da Qing Zhang. A service-oriented middleware for building context-aware services. *Journal of Network and Computer Applications*, 28(1):1 – 18, 2005.

[Gra12] Jefferson Graham. Ces 2012: Whirlpool shows off smart appliances. http://content.usatoday.com/communities/technologylive/post/2012/01/ces-2012-whirlpool-shows-off-smart-appliances/1#.UNHNY3eRb_m, Jan 2012. Accessed on 19 December 2012.

[GVRG+11] Julien Guibourdenche, Jacqueline Vacherand-Revel, Michèle Grosjean, Myriam Fréjus, and Yvon Haradji. Using multiple scores for transcribing the distributed activities of a family. In *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, CSCW '11, pages 637–640, New York, NY, USA, 2011. ACM.

[GWT+09] Tao Gu, Zhanqing Wu, Xianping Tao, Hung Keng Pung, and Jian Lu. epSICAR: An Emerging Patterns based approach to sequential, interleaved and Concurrent Activity Recognition. In *Proc. IEEE International Conference on Pervasive Computing and Communications PerCom 2009*, pages 1–9, 9–13 March 2009.

[HAR98] Y. HARADJI. L'analyse de l'activité et le modèle de tâches dans la spécification d'un système interactif. In *Ergonomie et informatique avancée (ERGO-IA '98), Actes du 6-ème colloque*, pages 174–185, Bayonne, France, 1998. Institut du logiciel et des systèmes. ISBN 2-9503134-9-3.

[HFS08] Tâm Huynh, Mario Fritz, and Bernt Schiele. Discovery of activity patterns using topic models. In *UbiComp '08: Proceedings of the 10th international conference on Ubiquitous computing*, pages 10–19, New York, NY, USA, 2008. ACM.

[HI04] K. Henricksen and J. Indulska. Modelling and using imperfect context information. pages 33 – 37, march 2004.

[HI06] Karen Henricksen and Jadwiga Indulska. Developing context-aware pervasive computing applications: Models and approach. *Journal of Pervasive and Mobile Computing*, 2(1):37–64, 2006.

[HIR02] Karen Henricksen, Jadwiga Indulska, and Andry Rakotonirainy. Modeling context information in pervasive computing systems. pages 167–180, 2002.

[HMEZ+05] Sumi Helal, William Mann, Hicham El-Zabadani, Jeffrey King, Youssef Kaddoura, and Erwin Jansen. The gator tech smart house: A programmable pervasive space. *Computer*, 38:50–60, 2005.

[HNM+09] Xin Hong, Chris Nugent, Maurice Mulvenna, Sally McClean, Bryan Scotney, and Steven Devlin. Evidential fusion of sensor data for activity recognition in smart homes. *Pervasive and Mobile Computing*, 5(3):236 – 252, 2009. Pervasive Health and Wellness Management.

[HPB+10] Jesse Hoey, Pascal Poupart, Axel von Bertoldi, Tammy Craig, Craig Boutilier, and Alex Mihailidis. Automated handwashing assistance for persons with dementia using video and a partially observable markov decision process. *Comput. Vis. Image Underst.*, 114(5):503–519, May 2010.

[Jen] Christian D. Jensen. Taxonomy of smart houses. http://www2.imm.dtu.dk/~cdj/SmartHouseWebSite/taxonomy.html. Accessed on April 9, 2012.

[KCC+09] Predrag Klasnja, Sunny Consolvo, Tanzeem Choudhury, Richard Beckwith, and Jeffrey Hightower. Exploring privacy concerns about personal sensing. In Hideyuki Tokuda, Michael Beigl, Adrian Friday, A. Brush, and Yoshito Tobe, editors, *Pervasive Computing*, volume 5538 of *Lecture Notes in Computer Science*, pages 176–183. Springer Berlin / Heidelberg, 2009.

[KHC10] Eunju Kim, Sumi Helal, and Diane Cook. Human activity recognition and pattern discovery. *IEEE Pervasive Computing*, 9:48–53, 2010.

[KPJ+08] Julie A. Kientz, Shwetak N. Patel, Brian Jones, Ed Price, Elizabeth D. Mynatt, and Gregory D. Abowd. The georgia tech aware home. In *CHI '08 extended abstracts on Human factors in computing systems*, CHI EA '08, pages 3675–3680, New York, NY, USA, 2008. ACM.

[LD11] Fei Li and Schahram Dustdar. Incorporating unsupervised learning in activity recognition. WS-11-04, 2011.

[MB97] John McCarthy and Sasa Buvac. Formalizing context (expanded notes), 1997.

[MO08] Arnaud Martin and Christophe Osswald. Generalized proportional conflict redistribution rule applied to sonar imagery and radar targets classification. *CoRR*, abs/0806.2008, 2008.

[Moz98] Michael C Mozer. The neural network house: An environment that adapts to its inhabitants. *Proc AAAI Spring Symp Intelligent Environments*, pages 110–114, 1998.

[Moz04] Michael C. Mozer. *Lessons from an Adaptive Home. In: Diane J. Cook and Sajal K. Das: Smart Environments*, chapter 12, pages 271–294. John Wiley & Sons, Inc., 2004.

[Mur00] Catherine K. Murphy. Combining belief functions when evidence conflicts. *Decision Support Systems*, 29(1):1 – 9, 2000.

[MYCD08] S. McKeever, J. Ye, L. Coyle, and S. Dobson. A multilayered uncertainty model for context aware systems. In *Adjunct Proceedings of the 6th International Conference on Pervasive Computing (Pervasive 2008): Late Breaking Result*, pages 1–4, Sydney, Australia, May 2008.

[MYCD09] Susan McKeever, Juan Ye, Lorcan Coyle, and Simon Dobson. Using dempster-shafer theory of evidence for situation inference. In *Proceedings of the 4th European conference on Smart sensing and context*, EuroSSC'09, pages 149–162, Berlin, Heidelberg, 2009. Springer-Verlag.

[NGS+01] Daniela Nicklas, Matthias Grossmann, Thomas Schwarz, Steffen Volz, and Bernhard Mitschang. A model-based, open architecture for mobile, spatially aware applications. 2121:117–135, 2001.

[NKG+15] YAMADA NAOHARU, SAKAMOTO KENJI, KUNITO GORO, ISODA YOSHINORI, YAMAZAKI KENICHI, and TANAKA SATOSHI. Applying ontology and probabilistic model to human activity recognition from surrounding things(mobile computing). *IPSJ Journal*, 48(8):2823–2834, 2007-08-15.

[OHG02] N. Oliver, E. Horvitz, and A. Garg. Layered representations for human activity recognition. pages 3 – 8, 2002.

[Oxy] Mit project oxygen. `http://oxygen.lcs.mit.edu/`. Accessed on 5 July 2012.

[Pad06] Amir Padovitz. *Context Management and Reasoning about Situations in Pervasive Computing*. PhD thesis, Monash University, Australia, 2006.

[PBC+03] Martha E. Pollack, Laura Brown, Dirk Colbry, Colleen E. McCarthy, Cheryl Orosz, Bart Peintner, Sailesh Ramakrishnan, and Ioannis Tsamardinos. Autominder: an intelligent cognitive orthotic system for people with memory impairment. *Robotics and Autonomous Systems*, 44(3-4):273 – 282, 2003. Best papers presented at IAS-7.

[PDW11] Bastien Pietropaoli, Michele Dominici, and Frédéric Weis. Multi-sensor data fusion within the belief functions framework - application to smart home services. In *Proc. ruSMART*, 2011.

[PDW12] Bastien Pietropaoli, Michele Dominici, and Frédéric Weis. Belief inference with timed evidence. In Thierry Denoeux and Marie-Hélène Masson, editors, *Belief Functions: Theory and Applications*, volume 164 of *Advances in Intelligent and Soft Computing*, pages 409–416. Springer Berlin / Heidelberg, 2012.

[PFH09] Germain Poizat, Myriam Fréjus, and Yvon Haradji. Analysis of activity in domestic settings for the design ubiquitous technologies. In *European Conference on Cognitive Ergonomics: Designing beyond the Product — Understanding Activity and User Experience in Ubiquitous Environments*, ECCE '09, pages 14:1–14:2, VTT, Finland, Finland, 2009. VTT Technical Research Centre of Finland.

[PRK+07] Shwetak N. Patel, Thomas Robertson, Julie A. Kientz, Matthew S. Reynolds, and Gregory D. Abowd. At the flick of a switch: detecting and classifying unique electrical events on the residential power line. In *Proceedings of the 9th international conference on Ubiquitous computing*, UbiComp '07, pages 271–288, Berlin, Heidelberg, 2007. Springer-Verlag.

[PTA06] Shwetak N. Patel, Khai N. Truong, and Gregory D. Abowd. PowerLine positioning: A practical sub-room-level indoor location system for domestic use. In *UbiComp 2006: Ubiquitous Computing*, volume 4206 of *Lecture Notes in Computer Science*, pages 441–458. Springer Berlin / Heidelberg, 2006.

[PZL06] A. Padovitz, A. Zaslavsky, and S. W. Loke. A unifying model for representing and reasoning about context under uncertainty. July 2006.

[RB09] Daniele Riboni and Claudio Bettini. Context-aware activity recognition through a combination of ontological and statistical reasoning. In Daqing Zhang, Marius Portmann, Ah-Hwee Tan, and Jadwiga Indulska, editors, *Ubiquitous Intelligence and Computing*, volume 5585 of *Lecture Notes in Computer Science*, pages 39–53. Springer Berlin / Heidelberg, 2009.

[SBCC01] F. Steenkeste, H. Bocquet, M. Chan, and E. Campo. La mise en place d'une technologie pour observer le comportement nocturne des personnes âgées en institution. *ITBM-RBM*, 22(1):25 – 30, 2001.

[SDFH09] Pascal Salembier, Julie Dugdale, Myriam Frejus, and Yvon Haradji. A descriptive model of contextual activities for the design of domestic situations. In *European Conference on Cognitive Ergonomics: Designing beyond the Product — Understanding Activity and User Experience in Ubiquitous Environments*, ECCE '09, pages 13:1–13:7, VTT, Finland, Finland, 2009. VTT Technical Research Centre of Finland.

[Sha76] G. Shafer. *A mathematical theory of evidence*, volume 1. Princeton university press Princeton, 1976.

[SK94] Philippe Smets and Robert Kennes. The transferable belief model. *Artificial Intelligence*, 66(2):191 – 234, 1994.

[SM4] Smart homes for all project (SM4ALL). http://www.sm4all-project.eu/. Accessed on April 9, 2012.

[Sme05] Philippe Smets. Decision making in the tbm: the necessity of the pignistic transformation. *Int. J. Approx. Reasoning*, 38(2):133–147, 2005.

[The03] J. Theureau. *Handbook of Cognitive Task Design*, chapter Course-of-Action Analysis and Course-of-Action-Centered Design, pages 55–81. Lawrence Erlbaum Associates, New Jersey, 2003.

[Wei93] Mark Weiser. Some computer science issues in ubiquitous computing. *Commun. ACM*, 36:75–84, July 1993.

[WGET08] Segev Wasserkrug, Avigdor Gal, Opher Etzion, and Yulia Turchin. Complex event processing over uncertain data. pages 253–264, 2008.

[WOC+07] Jianxin Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J.M. Rehg. A scalable approach to activity recognition based on object use. pages 1 –8, oct. 2007.

[WPC05] Danny Wyatt, Matthai Philipose, and Tanzeem Choudhury. Unsupervised activity recognition using automatically mined common sense. pages 21–27, 2005.

[Yag87] Ronald R. Yager. On the dempster-shafer framework and new combination rules. *Information Sciences*, 41(2):93 – 137, 1987.

[YCH05] G. Michael Youngblood, Diane J. Cook, and Lawrence B. Holder. A learning architecture for automating the intelligent environment. pages 1576–1581, 2005.

[YDM11] J. Ye, S. Dobson, and S. McKeever. Situation identification techniques in pervasive computing: A review. *Pervasive and Mobile Computing*, 2011.

[Zad65] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338 – 353, 1965.

[ZF10] M. Zouinar and M. Fréjus. *Ergonomie des produits et des services*, chapter Ergonomie Des Situations Domestiques : Nouveaux Défis, Nouvelles Opportunités. PUF, 2010.

Contents

List of Figures

Contributing to Energy Efficiency through a User-centered Smart Home

Smart homes are residences equipped with information and communication technologies that anticipate and respond to the needs of the occupants. Despite the numerous research and industrial efforts, today only few expensive smart homes have been built and sold. The reason behind this slow uptake is the technology-driven approach characterizing existing solutions.

The doctoral Thesis aims at demonstrating that a smart home can provide functionalities designed with a user-centered approach, taking into account ergonomic considerations about domestic activity and human cognition. This is achieved in collaboration with cognitive ergonomists, which help "minding the gap" between human context and machine-understandable context.

Using off-the-shelf and lightweight instrumentation (also minimizing privacy concerns), extending existing context modeling, reasoning and management tools and following the Ubiquitous Computing principles, the doctoral work led to the following achievements: *(i)* the inter-disciplinary design of suitable functionalities, in collaboration with cognitive ergonomists; *(ii)* the design of a context-aware system that captures and reasons about uncertain contextual information in a distributed fashion; *(ii)* the realization of a working prototype that demonstrates the provision of energy-saving and comfort-preserving functionalities.

Contribution à la gestion efficace de l'énergie dans le contexte d'une Maison Intelligente "Centrée Utilisateur"

Les *maisons intelligentes* sont des habitations équipées de technologies de l'information et de la communication qui anticipent et répondent aux besoins des occupants. Malgré les nombreux travaux et solutions existants, seulement peu d'exemplaires de maisons intelligentes ont été construits et vendus. La raison cachée derrière cette lente diffusion est l'orientation technologique des approches existantes.

Cette thèse de doctorat vise à démontrer qu'une maison intelligente peut fournir des fonctionnalités conçues avec une approche centrée utilisateur, en prenant en compte de considérations ergonomiques sur l'activité domestique et sur la cognition humaine. Ceci est réalisé en collaboration avec des ergonomes cognitifs, qui aident à "prendre garde" à l'écart entre le contexte humain et le contexte compréhensible par une machine.

En utilisant une instrumentation légère, qui minimise les problèmes d'acceptabilité et de protection de la vie privée, ce travail de thèse a mené aux contributions suivantes: *(i)* la conception interdisciplinaire de fonctionnalités adaptées, en collaboration avec des ergonomes cognitifs; *(ii)* la conception d'un système sensible au contexte qui capture et raisonne sur des informations contextuelles incertaines de façon distribuée; *(iii)* la réalisation d'un prototype qui démontre la fourniture de fonctionnalités qui réalisent des économies d'énergie, tout en préservant le confort des habitants.