DEPARTAMENTO DE TECNOLOGÍAS Y SISTEMAS DE LA INFORMACIÓN
ESCUELA SUPERIOR DE INFORMÁTICA
UNIVERSIDAD DE CASTILLA-LA MANCHA

TESIS DOCTORAL

AUTOMATIC SERVICE COMPOSITION BASED ON COMMON-SENSE REASONING FOR AMBIENT INTELLIGENCE

María José Santofimia Romero
Ingeniero Informático

Ciudad Real, 2011

DEPARTAMENTO DE TECNOLOGÍAS Y SISTEMAS DE LA INFORMACIÓN
ESCUELA SUPERIOR DE INFORMÁTICA
UNIVERSIDAD DE CASTILLA-LA MANCHA

TESIS DOCTORAL

AUTOMATIC SERVICE COMPOSITION BASED ON COMMON-SENSE REASONING FOR AMBIENT INTELLIGENCE

Autor:
María José Santofimia Romero
Ingeniero Informático

Directores:
Francisco Moya Fernández
Dr. Ingeniero de Telecomunicación
Universidad de Castilla-La Mancha

Juan Carlos López López
Dr. Ingeniero de Telecomunicación
Catedrático de Universidad
Universidad de Castilla-La Mancha

Ciudad Real, 2011

María José Santofimia Romero

Teléfono: (+34) 926 295300 ext. 3708
E-mail: mariajose.santofimia@uclm.es
Web site: `http://arco.esi.uclm.es/~mariajose.santofimia`

A mi familia

Resumen

La Inteligencia Ambiental, plasmada en lo que se conoce como *entornos inteligentes*, está fundamentalmente orientada a simplificar la vida diaria de las personas; para ello, este paradigma hace recaer en el entorno la responsabilidad de prever, identificar y satisfacer necesidades concretas.

Estos entornos inteligentes, por ser autosuficientes o autónomos, son el contexto ideal donde personas de la tercera edad o personas con algún tipo de discapacidad o enfermedad, podrían desarrollar su vida con mayor normalidad y autonomía. El entorno, consciente de las limitaciones de estas personas, supervisaría el contexto reaccionando y supliendo de manera oportuna dichas limitaciones. Sin embargo, éstos no son los único ámbitos en los que este paradigma puede hacer grandes aportaciones. Cualquier contexto susceptible de ser monitorizado y controlado mediante dispositivos electrónicos puede ser automatizado, desde la óptica de la Inteligencia Ambiental, para trabajar de manera autónoma y no supervisada. Otro ejemplo de entorno inteligente, fuera del ámbito del hogar, son los edificios o recintos cuya vigilancia se basa en la información captada por sensores y en las decisiones que, derivadas de la interpretación de esa información, están destinadas a mantener las condiciones de seguirad del recinto.

Independientemente del ámbito de aplicación, existen una serie de requisitos o necesidades que son comunes a todos ellos. Las principales características de los entornos inteligentes son su autonomía y capacidad de adaptación a los cambios en el contexto. De esta manera, ambientes inteligentes son todos aquellos entornos que puedan beneficiarse de un sistema de supervisión que comprenda lo que ocurre a su alrededor y que pueda actuar en consecuencia. Llevar a cabo esas tareas de manera sutil y casi imperceptible para las personas son dos de los grandes retos de este paradigma.

En el desarrollo de sistemas para entornos inteligentes, la generación automática de respuestas a las necesidades del entorno es el cuello de botella que está ralentizando la consecución de entornos realmente inteligentes. Una vez identificada y analizada esta problemática, el resultado de este trabajo de tesis pretende aportar soluciones a la misma. En esta tesis se ha trabajado en la composición automática de servicios como mecanismo para articular esas respuestas. Sin embargo, resulta evidente de que la articulación de respuestas mediante la composición de servicios debe estar fundamentanda no sólo en un profundo entendimiento de la situación contextual, sino también en el conocimiento general que determina cómo funcionan las cosas, lo que se conoce como *sentido común*.

De manera resumida, podemos decir que este trabajo de tesis estará orientado a comprender, entender y planificar una solución que dé respuesta a necesidades emergentes en entornos inteligentes. Adoptar un enfoque basado en “sentido común” parece una de las alternativas más coherentes para abordar esta tarea. Así, lo que persigue este trabajo es desarrollar sistemas capaces de imitar el comportamiento que las personas tendrían ante situaciones similares, entendiendo por sentido común ese conocimiento global que las personas poseemos acerca de cómo funciona el mundo. Conseguir

transmitir ese conocimiento será, sin duda, una de las piezas claves para la consecución de sistemas inteligentes.

Abstract

The Ambient Intelligence paradigm is mainly devoted to make people life easier, by means of the so-called *smart spaces*. To this end, the Ambient Intelligence paradigm rest upon the environment the responsibility to foresee, identify, and satisfy the arising need or requirements.

These smart spaces are thought to be self-sufficient and autonomous, and therefore they are the ideal contexts for elder people, or those with some degree of disability. The fact that the context is aware of the activity in which the person is engaged enables the environment to simplify and ease the achievement of such activity, since the environment is also aware of the limitations of those people. Nevertheless, these are not the only contexts in which Ambient Intelligence might provide a great help. Any context capable of being monitored and supervised by means of electronic devices can be automated, from the optic of the Ambient Intelligence, in order to work in an autonomous and manner minimizing human intervention. Besides home contexts, Ambient Intelligence can also be applied to the surveillance of buildings, on the basis of the information gathered from the sensing devices deployed in the context. These buildings are also expected to make decisions that, grounded in the information gathered from the environment and once it is interpreted, are intended to maintain the security conditions of the building.

Independently of the application contexts, there exists some common requirements or needs that are common to all of those intelligent environment. Basically, those environments are mainly characterized by their autonomy and self-adaption to the context changes. Ambient Intelligence environments are therefore all those that can benefit from a supervision system capable of understanding the ongoing situation and that can consequently adopt the most appropriate behavioral response. Accomplishing these task in a seamless and imperceptible manner are two of the main challenges of this paradigm.

The development of Ambient Intelligence systems poses an additional challenge, as it is the automatic response generation, according to the environmental needs. This is the bottleneck of the Ambient Intelligence systems which is preventing these smart spaces from being achieved. This thesis is mainly concerned about this problem, and it is intended to devise a solution to this hurdle. Automatic service composition is advocated here as the enabling strategy to articulate environmental behavioral responses as though they were composite services. In order to do this, it is necessary to both, provide support for a deep understanding of the contextual situation and to hold the general knowledge that dictates how *the world works*, the so-called *common sense*.

Summarily, this thesis work is intended to devise an architectural solution to understand and plan the most appropriate response to the environmental arising needs for Ambient Intelligence. Adopting a common-sense approach seems to be the most plausible solution to tackle this endeavor. This thesis pretends to provide a solution for developing systems capable of imitating human behavior.

Contents

List of Tables

List of Figures

Agradecimientos

Ha sido mucha la gente que me ha ayudado a llegar hasta aquí y por eso me resulta muy difícil tener que resumir tanta gratitud en tan poco espacio.

En primer lugar, esta tesis no hubiera sido posible sin el apoyo incondicional de mis directores de tesis. Estoy especialmente agradecida a Juan Carlos, porque de su dedicación, esfuerzo y compromiso he aprendido que la meta última ha de ser siempre la búsqueda de la excelencia, “*there is always room for improvement*”. A Paco, por confiar tanto en mi y por todo lo que he aprendido de él. Finalmente, y aunque como directores de tesis sólo figuren dos, ha habido un tercero, que a la sombra, me ha enseñado todo lo que se necesita saber para sobrevivir a una tesis. Xavi, más allá de “*for your love and support*”, te estoy muy agradecida porque has sido el que más horas le ha dedicado a esta tesis, porque te has encargado de la cuestión gráfica y porque has revisado hasta la última referencia de esta tesis, pero sobre todo porque tuyo es el mérito de que esta tesis haya llegado a su fin. Eres una gran persona y un gran compañero.

Tengo que agradecer también a todos los miembros del grupo ARCO, pero de manera especial a Félix por darme la oportunidad de estar aquí y por ser el que, en primer lugar, planteó el problema de la composición automática de servicios. Además, una tesis son muchas horas de trabajo y por eso tengo que agradecer profundamente a Jesús porque como compañero de despacho, con sus consejos y bromas, ha hecho más llevadero el camino hasta ver terminada esta tesis. A Julio, porque su actitud positiva ante la vida es para mi un ejemplo a seguir. A Fernando, no sólo por ser el que cada mañana pronuncia la palabra mágica “*¿café?*”, sino también, porque de su entusiasmo y dedicación por la docencia he aprendido mucho. Finalmente, a Juan Pablo y a su Mónico Sánchez, gracias a los cuales entré en el “*maravilloso mundo*” de la investigación.

Disfruto especialmente dedicando y agradeciendo este trabajo a mi familia, y especialmente a mis padres. A mi madre, porque me ha enseñado que con coraje y esfuerzo se consigue todo en esta vida. Mamá, te estaré eternamente agradecida por lo mucho que has sacrificado para que yo haya podido llegar hasta aquí. Ahora te toca a ti, estoy segura de que llegarás a ser una gran enfermera. A mi padre, porque hay que ir muchas veces a Alemania, pasar mucho frío y “comer de *cajoncillo”* muchas veces para que una hija “*se haga una señora doctora ingeniera*”. A mi hermana Irene, porque aunque ella no lo sepa, también me ha ayudado mucho a llegar hasta aquí. A Veredas Felicia, por ser la alegría de la familia. Finalmente, también quiero agradecer a mis abuelos, tíos y primos, por hacer de esta tesis un miembro más de la familia.

Los amigos van y vienen, pero hay algunos que siempre están ahí. Merche, muchas gracias por lo mucho que me has animado en mi carrera como “*investigadora*”.

Last but not least, I wish to express my gratitude to Scott E. Fahlman for helping me so much in the accomplishment of this thesis.

Part I

Preliminaries

Chapter 1

Introduction

> **Common sense is the collection of prejudices acquired by age eighteen.**
>
> – *Albert Einstein*

Summary – *The first chapter of this document is devoted to introducing and justifying the interest of the problem addressed by this thesis. This chapter also outlines the most relevant background related to this research work. Finally, the main contributions of this work are presented along with the structure of the document.*

1.1 Introduction

Progress made in mobile and wireless technology led the computing science community to postulate the advent of a new computing era in which seamless access to services and information were going to be available at anytime and from anywhere. The first attempt to theorize this new era of computing was due to Mark Weiser, who in [148] hypothesized this novel interaction paradigm under the name of Ubiquitous Computing. The Ubiquitous Computing paradigm envisioned by Mark Weiser predicted an upcoming time of invisible electronic devices, in which such devices appeared to be merged with the background and the interaction was performed unconsciously, focusing on what people was trying to do rather than on how they had to do it.

Over the past ten years, the ideas of Mark Weiser about the invisible computer have been growing in interest and important research efforts have been dedicated in that direction. As a consequence, the great concern in this field led the IST Advisory Group to coin the term "*Ambient Intelligence*" [35] for referring to those envisioned environments, where those surrounding, intelligent and intuitive devices are capable of recognizing and responding to the arising needs and requirements. Roughly, it can be said that the Ambient Intelligence concept extends the Ubiquitous Computing paradigm by enhancing devices with the required intelligence to support humans in their everyday activities.

Additionally, if the Ambient Intelligence concept was to be summarized using just three words they would be *seamless*, *unobtrusive* and *invisible*. Nonetheless, for a further description of the Ambient Intelligence principles it is essential to analyze the scenarios depicted in [35]. Those scenarios were supposed to provide a likely glimpse of how computer interactions were expected to be in 2010. However, now that 2010 has passed, it is an unquestionable fact that there is still a gap between the scenarios envisioned by the European Commission and what has been achieved so far by the scientific community.

This gap poses an urgent need to determine the cause that is preventing Ambient Intelligence from becoming a reality. In this regard, the arising question is why those or similar scenarios have not yet been reproduced. Implicitly, this question asks whether it is due to the available technology or to the poor achievements in interdependent fields. The answer to this question will determine the direction in which efforts need to be expended in order to overcome the gap referred to.

At first glance, it seems obvious that technology has properly evolved towards pervasiveness. It is a fact that electronic devices are gaining in computational resources while at the same time they are decreasing in size and cost. These features are in consonance with the scenarios described by European Commission. However, aspects such as the lack of standards, heterogeneity, or device dynamism have posed important challenges to an efficient and effective use of the services provided by the available technology. Despite these drawbacks, it can be advanced that different approaches have been formulated to address such challenges with different levels of success.

To the best of our knowledge the weak point of the Ambient Intelligence paradigm is rooted neither in technology nor in how it is being exploited. The core problem that prevents Ambient Intelligence from being a reality is found in the implementation of the reasoning capabilities that should lead the environment to respond in accordance with the ongoing situation. In this respect, the leading cause of the aforementioned gap suggests that research efforts should be addressed to overcoming the over-simplistic implementations of cognitive and understanding capabilities that have characterized the proposals presented to date.

For the sake of disclosure, the problem of building systems for Ambient Intelligence should therefore be addressed from the same perspective as building intelligent systems. Note that in contrast to expert systems, in which specialized knowledge about a specific topic is required, intelligent systems should count on a great deal of knowledge that support them in dealing with a wide variety of problems. Not only general knowledge is required but also the capacity to reason using that knowledge. Unfortunately, ineffective means to understand and reason in ongoing situations lead to unsuccessful systems for Ambient Intelligence. Those systems are expected to exhibit an autonomous and intelligent behavior by understanding and reacting to the activities that are taking place. It is therefore essential to this sort of system to be built upon an application that provides them with a proper means to do both, manage domain information and to generate appropriate responses depending on that information.

Understanding the activities that are being carried out in an Ambient Intelligence environment involves the interpretation of the events that are being captured by the electronic devices deployed in it. Apparently, this activity looks like a trivial and simple task. At least, humans do not seem to have a problem in inferring on-going situations given the occurrence of a set of facts. For example, if the light is off, the projector is on, and there are people in the room, it can be easily inferred that a presentation is likely to be going on. However, and in contrast to what it might be expected, computers struggle to succeed in such endeavors. Cognitive and understanding capabilities are the mechanisms that enable humans to succeed in identifying and recognizing situations out of a set of events taking place.

In order to identify the reason that underlies the poor success of computing approaches in cognition and understanding we need to analyze the features that distinguish people from computers, considering the cognitive and understanding capabilities. The main difference is rooted in the vast amount of knowledge held by humans as well as in the associated mechanisms that support an effective use of such knowledge. It is important to highlight that the demand for cognitive and understanding capabilities responds to an indirect claim for self-management, pro-activeness, dynamism and goal-driven behavioral features. Far from being trivial matters, these demands constitute some of the most challenging requirements that have to be tackled when developing systems for Ambient Intelligence [113].

As already stated above, an additional difficulty in developing Ambient Intelligence systems is that of having to cope with the wide range of device technologies that populate these environments. Device capabilities to cooperate and interact among themselves are inherent to the Ambient Intelligence paradigm. However, it cannot be forgotten that low-level details concerning communication protocols, operating systems, or programming languages, etc., are overlooked aspects that hamper seamless cooperation among devices. This concern has been addressed, with different levels of success, by a wide variety of techniques, such as web services [71], middleware [55] [16], dynamic reconfiguration [17], agents [18], context modeling or reasoning approaches [116] [103] among the most relevant ones.

Along with the device heterogeneity feature, it is also worth remembering that device continuity cannot be taken for granted. On the contrary, electronic devices are constantly appearing and vanishing from the scene. For example, the appearance of a person that holds a cell phone should also means that a new device is being offered new services. Whenever the person abandons the supervised space, the cell phone stops providing services. The main implication of this discontinuity in the service availability is that environmental responses cannot be statically pre-coded as service instantiation recipes. On the contrary, systems for Ambient Intelligence are expected to devise in an ad-hoc manner the most appropriate means to undertake a specific purpose.

The purpose of these introductory paragraphs has been to provide the reader with a clear idea of the unresolved challenges that are preventing Ambient Intelligence from being a reality. The conclusion that can be drawn from this initial analysis, which also serves as the initial working hypothesis of this thesis, is that the autonomy expected from Ambient Intelligence systems can only be achieved by leveraging cognitive and behavioral human-like capabilities. In this respect, in the late 60's McCarthy postulated in [89] that only by endowing programs with common sense could they be able to achieve the intelligence sought. Doug Lenat in [80] also pointed in the same direction, highlighting the importance that common-sense knowledge has when dealing with and reacting to novel situations. The agreement on the key role that common sense plays in building intelligent systems is one of the axiomatic facts of this thesis. Efforts are therefore addressed to enact common-sense knowledge and reasoning capabilities as a prerequisite to support the cognitive and understanding processes demanded by Ambient Intelligence.

Additionally, understanding processes are also dependent on the capacity to perform causal explanations of the facts, which come in the shape of sensor values, retrieved from the environment. According to Davidson [30], the reason that motivates an action also rationalizes it, and therefore, this thesis is also grounded in the working hypothesis that an appropriate philosophy of actions should be adopted so as to support the aforementioned cognitive and understanding demands.

Contrary to the current practices found in the literature review, Ambient Intelligence systems should be grounded in a philosophy of actions that does not respond to the convenience of the implemented approach, but rather it does respond to the conclusions drawn from the philosophical doctrine, regarding actions, events, and reasons. The models surveyed in [121] bring to light the different characterization of the notion of event, action, or service depending on the targeted context. This thesis is rooted in the belief that the semantic model should be unique and grounded in a rational explanation of actions and events.

Summarizing the aforementioned axioms and working hypotheses, the problem of building systems for Ambient Intelligence has to be tackled from both the cognitive and behavioral perspectives. Additionally, the proposed solution should be built upon the basis of a semantic model for actions and events, and the mechanisms to resemble human-like common-sense knowledge and reasoning capabilities.

From the cognitive perspective, the problem can be addressed as an understanding problem. Comprehending a situation that takes place in a context might involve, for example, the inference of implicit, non-deterministic or delayed effects. A delayed effect of turning on a tap in a kitchen sink whose plug is in place, will be a water overflow. From a behavioral perspective, the problem can be addressed as a planning problem of deciding what action to take in certain given circumstances. A common-sense strategy for planning and understanding, such as that presented in [150] would, therefore, appear to be the most compelling approach towards emulating the human-like rationality and reasoning capability.

Finally, the last challenge that should be faced when building systems for Ambient Intelligence is that of addressing novelty. It has to be noted that unexpected situations, which in words of Doug Lenat et al. [80] suppose the bottleneck of intelligent systems, are the most common situations found in Ambient Intelligence contexts. The way in which people react to these unexpected situations provides an idea of the direction in which efforts should be addressed. Generally, when facing new situations, people tend to establish some similarities with past experiences, or resort to their general knowledge of how things work, the so-called common-sense knowledge, or even look for advice in books. Whatever the case may be, it is axiomatic for this thesis that only Ambient Intelligence systems will be sufficiently flexible to support the scenarios envisioned in [35] when common-sense reasoning starts being considered as a structural part of such systems. In this sense, the associated working hypothesis is that understanding and modeling common-sense reasoning, in such a way that it can be automatically performed, is a key challenge that, once achieved, would allow systems for Ambient Intelligence to be indeed intelligent.

At this point, the main axioms and working hypotheses of this work have been roughly outlined, providing an overall view of the main challenges faced when building systems for Ambient Intelligence. The following sections will address with higher levels of detail, the fundamental problems that have motivated this work. The third section will present the aims and objectives of this thesis. While the main contributions are presented in section four. Finally, the last section presents a brief description of the content addressed in each of the consecutive chapters that follow this chapter.

1.2 Motivation

The previous section has pointed out some of the main challenges that need to be addressed when building systems for Ambient Intelligence. These challenges are also considered to be part of the main concerns that have motivated this thesis. However, for the sake of precision, this section simply synthesizes these general aspects by providing a list of intentions, which will be addressed with more detail in the following chapters. These intentions establish the directions towards which efforts need to be directed in order to achieve the ultimate goal of building systems for Ambient Intelligence.

In this regard, the main peculiarity which distinguishes Ambient Intelligence from other fields of knowledge is the interdisciplinarity of the challenges faced. This interdisciplinarity therefore involves specialization in a wide variety of technologies, thus turning the task of building comprehensive approaches to Ambient Intelligence into a very complex endeavor that requires knowledge of a wide range of technologies.

Augusto et al. enumerate in [8] some of the main technologies that are encompassed in Ambient Intelligence. As stated by these authors, building systems for Ambient Intelligence is a task that requires a certain level of expertise in fields such as Sensor Networks, Artificial Intelligence, or Multi-Agent Systems among others. The adoption of an approach based on a single technology does not suffice to tackle the endeavor of enhancing Ambient Intelligence systems with capabilities to behave in a self-managed, pro-active, dynamic and goal-driven fashion. On the contrary, a comprehensive approach is demanded by the need to address the different problems that arise in this regard. However, the highly demanding task of devising such a comprehensive solution cannot always be afforded by small research entities. As will be described in the next chapter, the main contributions to this field come from big research consortia or from European funded projects. This situation poses the need to make assumptions or to overlook relevant aspects when trying to make a contribution in this field.

It is therefore futile for this thesis to attempt to face all these challenges from scratch, although at the same time, this thesis is indeed engaged in minimizing the number of assumptions required. This endeavor can be feasibly undertaken since this thesis is encompassed in the comprehensive approach carried out by the ARCO Research Group, at the University of Castilla-La Mancha. Under the direction of this Research Group, a divide-and-conquer approach has been adopted towards Ambient Intelligence. It has been necessary to identify the different disciplines that include the challenges that, to date, remain unsolved, and address them individually. As a result, this thesis rather than having to deal with all these challenges from scratch, can focus on a specific set of them, and ground the proposed approach in the research and technological solutions provided by previous projects and theses developed within the group that have contributed to the state of the art.

The multidisciplinary fields in which the ARCO Research Group has specialized entitles it to undertake a comprehensive approach to building Ambient Intelligence systems, in which three main layers can be identified. The inner layer deals with technological issues, the second layer abstracts the communication aspects, and finally, the third layer focuses on orchestrating the underlying layers. This third layer is responsible for enabling the overall system to resemble cognitive and behavioral human-like capabilities.

In this respect, the first two layers correspond with two theses completed within the activities of the research group. The first layer has been devoted to managing the different technologies that might appear in an Ambient Intelligence environment. In this regard, the lack of standards homogenizing the access to the deployed devices, demands some additional mechanisms that automate functions such as the device and service discovery. It is also necessary to apply some adaption mechanisms in order to conceal the technological details to the upper layers, providing a standardized manner for using the services provided. These challenges have been addressed in the thesis work of Dr. D. Villa [144], which has therefore contributed with an effective solution to address the device dynamism and heterogeneity problem that characterize Ambient Intelligence environments. The second layer deals with the communication issues, in seeking to reproduce a *Place & Play* philosophy, that serves to overlook the technology-dependent details. These challenges have been addressed in the thesis of Dr. F. J. Villanueva [146] that has contributed to the state of the art with an architectural approach to cater for the Ambient Intelligence deployment requirements.

The present thesis is framed in the results and conclusions drawn from these previous works. In this respect, the contributions made by the aforementioned thesis lay the foundations for the integration and communication of sensor and actuator networks, proposing an efficient solution to address the dynamism and heterogeneity that characterize such contexts. Nevertheless, the Ambient Intelligence environment is not yet capable of autonomously understanding and reacting the unforeseen situations, and that is the gap that this thesis has intended to fill.

Traditionally, Ambient Intelligence solutions have been mainly concerned with anticipating user actions and needs. For example, resorting to the first scenarios described in [35], the following question is therefore stated: What would be the system response if, on her arrival, Maria twists her ankle. Would the Ambient Intelligence system be capable of recognizing the possible injury in

Maria's ankle as an impediment to driving a car? Moreover, would the system be intelligent enough to infer that Maria should be asked about her preference to rent a car or take a taxi? These are some of the situations that more likely have not been considered in advance, and therefore, pre-coded responses have not been provided to the system. However, true intelligence is found in this sort of situation and not in implementing recipe-like solutions that, at some point, could also be disturbing or annoying.

Ambient Intelligence however, is not just about behavioral pattern matching, but it is also about the wise supervision of the environment, by satisfying those unforeseen requirements or needs by means of rational decisions in the same way a human would do. This thesis goes a step forward towards Ambient Intelligence and focuses on addressing the problem of dealing with unforeseen situations. An essential demand for this endeavor is to count on the appropriate cognitive capabilities to understand the environment. Additionally, once ongoing situations have been understood, it is also essential to be capable of rationally reacting to the recognized situation, as a person would do under the same circumstances. It seems unavoidable to implement cognitive and behavioral capabilities as constituent parts of the proposed solution to achieve Ambient Intelligence systems that are indeed intelligent.

However, these two capabilities are not only relevant when unexpected situations arise, but also when available means cannot be determined in advance. It has to be noticed that the main impact of the device dynamism is found in the inability to foresee the available means or devices that can be employed in implementing environmental reactions. For example, establishing a similarity with a common situation faced by people in their daily life, when opening a parcel, the most sensible approach is to use a pair of scissors to tearing up the tape. However, it might happen that the specific tool for this task, normally a pair of scissors, is not near at hand. Then, the natural reaction is to look for a similar tool or combination of tools that produce the same effects. More likely, the expected reaction would be to look for an object sharp enough to tear up the tape of the parcel. Basically, this behavior consists in working out an additional mechanism to undertake a task that is normally performed by means of a tool or a mechanism that is unavailable at that moment in time.

Therefore, what people do in this sort of situations is what Ambient Intelligence systems should do in the pursuit of the capability to deal with unpredicted situations. Since services are considered to be the tools that Ambient Intelligence systems have at their disposal, the only way of figuring out an alternative way of accomplishing a task for which there is not any specific service is by composing the available ones into a new service that emulates the desired functionality. Again, this composition task has to be undertaken in an autonomously and self-sufficient way without requesting user intervention, and for this reason it is referred as automatic service composition.

Automatically composing services in dynamic contexts is a task that entails a high degree of adaptability and understanding capabilities. Note that this task is mainly intended to devise the most appropriate composition of services that can cater for the required functionality, while at the same time, being consistent with the new scenarios and events that are concurrently taking place in the surrounding environment.

In this sense, the main contribution of automatic service composition stems from the versatility of the system responses that can be therefore undertaken, on the basis of the available services. In other words, the main contribution of automatic service composition to the Ambient Intelligence field rests on the fact that it enables systems to implement the most appropriate responses to ongoing situations even though the services assigned to this end are not available.

Apart from the understanding capabilities that are implicit to the automatic service composition, this task also needs to implement additional competences that enable it to proceed in the same way a person would do. Therefore, for the automatic service composition to succeed in responding to arising functionality demands, it is necessary to imitate the human responses by gaining knowledge of the context, establishing goals and employing reasoning capabilities in order to achieve the targeted functionality.

It is necessary to pay closer attention to at the way humans rationally decide how to proceed in unforeseen situations. The most characteristic feature of humans that distinguish them from other species is the collection of beliefs and reasoning capabilities known as *common sense*. As has been outlined so far in this document, this thesis adopts a common-sense approach to the problem of leveraging cognitive and behavioral capabilities for Ambient Intelligence. Efforts should be primarily intended to replicate and automate common-sense capabilities, so as to afterwards address the problem of resembling cognitive and behavioral capabilities.

For the sake of clarity, the matters that have been set out in this section can be summarized as the following set of items motivating this thesis:

1. Ambient Intelligence is a multidisciplinary paradigm that requires a comprehensive solution to address the wide variety of challenges that arise when building systems for Ambient Intelligence.

2. The previous work undertaken within the activities of the ARCO Research Group provides the basis to tackle the comprehensive endeavor of building Ambient Intelligence systems that are indeed intelligent.

3. The main shortcoming of current approaches is found in having to deal with unexpected situations. Novelty is therefore the main threat for Ambient Intelligence.

4. Common sense is one of the features that distinguishes humans from the rest of the animals. It is also responsible for the human ability to deal with novel situations. The lack of common-sense knowledge and reasoning capabilities is therefore responsible for the poor response to unforeseen situation that characterizes Ambient Intelligence.

5. Before a decision can be made about how to properly react to an ongoing situation it is necessary to understand and to recognize such situation. Contextual information is acquired from the sensor devices deployed in the context. This information should be therefore translated into semantic facts of the situation in progress.

6. In the same way people drive their behavior motivated by their beliefs, goals and intentions, Ambient Intelligence systems should be provided with the same sort of information if they are expected to behave in the same way people do.

7. Ambient Intelligence systems should also count on a set of mechanisms that enable them to undertake the responses that drive these systems to the targeted goals.

8. Ambient Intelligence systems have to be flexible enough to tackle any situation that might arise in a supervised context. Behavioral recipes do not suffice to cope with the unpredictable number of different situations.

1.3 Aims and objectives

Given that the previous section has enumerated the challenges that motivate this thesis, this section is therefore entitled to materialized such challenges into a concrete list of aims and objectives.

Generally speaking, the main objective of this thesis is to leverage those capabilities that enable Ambient Intelligent systems to effectively address unforeseen situation in a rational and human-like manner. Implicitly, this unspecific aim encompasses a set of more specific sub-objectives, the ones towards which efforts should be directed in order to achieve the ultimate goal of providing Ambient Intelligence indeed intelligent.

The comprehensive approach, that has been mentioned so many times in the previous section, refers to the fact that rather than having to face a single-objective endeavor, building Ambient Intelligence systems involves having to tackle several lower entity aims. In a figurative sense, these lower entity objectives can be seen depicted in a tree-like structure. At the top of the structure is the more general aim, as it is the intention of providing Ambient Intelligence with the capability to react to any of the situations that might arise in the context, just on the basis of a set of goals to be achieved or maintained.

Descending through that tree, the top goal can be split into two sub-objectives of a smaller entity. On the one hand, it is necessary to understand the activities that are being carried out in the surrounding context. Obviously, the data retrieved from the context need to be properly interpreted so as to infer the ongoing situation simply from the captured and sensed values. On the other hand, whenever any of the ongoing situations lead the context away from any of the desired goals, then some decisions should be made in order to return the context to the desired state. Basically, these two sub-goals capture the need for Ambient Intelligence to analyze the ongoing situation and to makes the right decision, in response to an emerging need or a goal deviation.

These sub-goals are therefore strongly connected to the processes in charge of both capturing and understanding events. The process involved in such activities therefore comprises the set of objectives considered at the third level of that tree-like goal structure. At this level, objectives are getting more specific. Basically, in order to process events it is necessary to count on a complete event model. This model poses the basis for capturing environmental events and mapping them into the systems, in seeking for a higher level interpretation. Additionally, a thorough description of world events needs to be provided, as if they were event patterns.

After the identification of the ongoing situation, new requirements or needs might arise that prompt the Ambient Intelligence system to make a decision on the basis of the environmental goals. This decision needs to be carried out by means of the services that are available at that specific moment. This layer is mainly nourished from the processes involved in the automatic service composition. Basically, this sub-objective can be summarized as achieving automatic service composition. An additional implication in achieving automatic service composition is that the available services need to work together, in an orchestrated and seamless manner, and without requiring people to make any additional configuration task.

Finally, at the bottom layer, the sub-goals upon which the aforementioned ones are supported involve the achievement of common-sense knowledge and reasoning capabilities. Notice that raw pattern matching does not suffice to map sensed data into ongoing situations, nor does to overcome the lack of services when it comes to undertake recipe-like responses. In this sense, the analysis of the human-like cognitive and behavioral capabilities reveals the need to leverage common-sense in Ambient Intelligence systems if they are intended to resemble human-like behavior. Therefore, the bottom layer is aimed at gathering the knowledge that describes how the world works, the so-called common-sense knowledge. Additionally, inference and reasoning mechanisms also need to be provided in order to be able to derive new information out of the existing knowledge and the information about the ongoing situation.

At the risk of being too pretentious and ambitious, this work pretends to propose a comprehensive solution to Ambient Intelligence that considers common sense as a constituent part of it. For the sake of precision, the aforementioned goals are detailed in the following specific goals:

1. To capture and model the specific common-sense knowledge that is directly involved in Ambient Intelligence. Special attention should be paid to the knowledge involved in modeling actions and events. Note that this aim is basically constrained to the knowledge that is somehow related to Ambient Intelligence. It is unfeasible to undertake the task of modeling the large amount of knowledge that people normally hold.

2. To support the context modeling and reasoning. The sensed information should be translated into semantic facts. These facts should be considered along with the knowledge that describes how things work in order to draw further knowledge about the ongoing situation or circumstances.

3. To devise a model of agency for Ambient Intelligence systems. This model is intended to provide the beliefs and goals that determine the direction towards which the Ambient Intelligence system responses should be directed.

4. To provide an automatic service composition mechanism that supports the behavioral response implementation of the Ambient Intelligence system. On the one hand, the system goals and beliefs determine the best way to proceed while on the other hand the automatic service composition mechanism enables the system to undertake the devised responses.

5. To leverage common-sense knowledge and reasoning capabilities in the automatic service composition. Some common-sense aspects concerning the effects of events are particularly relevant for a deep understanding of the ongoing situation, as well as for the implementation of the devised behavioral response.

6. To investigate the most appropriate means to capture and manage the context events as a requirement for context understanding.

The first items should be clearly undertaken from the perspective of a knowledge engineering task; in order to interpret contextual information, it is necessary to structure and organize the gathered knowledge as well as to semantically enhance it. On the basis of such semantically enhanced facts, new knowledge can be inferred or deduced. It has to be noted that this aim poses the demand for an adequate language and a reasoning system.

The semantic enhancement of the contextual information might not be enough to understand certain aspects. Understanding a situation that takes place in a context might involve, for example, the inference of implicit, non-deterministic or delayed effects.

The third point is closely related to the agent theory. It has to be remarked that the ability to replicate human reasoning remains one of the main objectives in the Artificial Intelligence field. In this regard, intelligent agents have came to be a powerful solution to this issue, and therefore, it might be an appropriate starting point from which to tackle this goal.

The fourth item concerns the way the desired state can be achieved given the available set of services. The planning theory contributes not only towards providing articulated responses by means of service composition, but also towards supporting the decision making of agents that exhibit a goal-oriented behavior. Providing a fully automated solution to compose services seems to be the most appropriate way to enrich systems with the autonomy demanded by Ambient Intelligence. The problem of supporting automatic service composition can be therefore formulated as an action planning problem upon the basis of common-sense knowledge.

1.4 Structure of the thesis

Chapter 2 provides a revision of the existing works and projects in the field Ambient Intelligence. However, and for the sake of clarity, the synergistic fields of knowledge that surround Ambient Intelligence are not analyzed there. On the contrary, these topics are being individually addressed in the following chapters.

Chapter 3 is the first of the part dedicated to addressing the *understanding* problem. This chapter focuses on setting the tenets of the common-sense theory, starting from a thorough revision of the key issues for common-sense reasoning. Additionally, this chapter is also devoted to analyze the contributions that common sense brings to Ambient Intelligence.

Chapter 4 deals with modeling and reasoning about context, also from the perspective of the common-sense tenets. This chapter presents a triple approach to modeling, based on a triple conception of the context, as known: the syntax, the semantics, and the pragmatics. An appropriate modeling language, that also considers these triple conceptions, should be provided in order to model and reason about context.

Chapter 5 is intended to understand the context of ongoing situations by understanding the actions and events that take place in the context, and by mapping them to specific situations. The correct identification of the ongoing situation is essential for devising the appropriate behavioral response. This chapter is also dedicated to highlighting the role that common sense plays in understanding the high level situation that is taking place, on the basis of the given events.

The following part is dedicated to revising the proposed approach to support the behavioral response implementation. This part is divided into two chapters; Chapter 6 is devoted to outlining the planning strategy in charge of devising the most appropriate solution given the ongoing situation and the goals to be achieved or maintained. Chapter 7 presents the planning strategy that is provided in order to implement the devised solution, presented in Chapter 6.

The last part of this document is intended to evaluate and validate the conclusions drawn from this thesis. Chapter 8 is intended to assess the performance of the given prototype. It has to be noticed that the previous chapters have been devising a set of benchmark problems that better fit the requirements of the evaluated scenarios. For that reason, those benchmark problems are undertaken and evaluated in this section.

Finally, Chapter 9 presents the conclusions and future lines of work drawn from this thesis. Previous chapters have been advancing the interim conclusions drawn from the studies undertaken in each particular chapter. These interim conclusions are summarized and synthesized in this chapter.

Chapter 2

State of the Art in Ambient Intelligence

“Ambient Intelligence is more than just a question of embedding technology into objects. It involves human culture in its broadest sense - universal desires; complex social relationships; different value systems; individual likes and dislikes; the sustainability of economic and natural ecosystems; and codes of ethics, conduct and communication, both in civil society and in business.”

– *Stefano Marzano*

Summary – *This chapter is devoted to surveying the most relevant and comprehensive solutions that, to date, have been proposed to answer the demands arose when building systems for Ambient Intelligence. Among those demands, special attention is paid to determining modeling and, how they also comply with the flexibility requirements through the implementation of any of the different mechanisms for service composition.*

2.1 Introduction

The vast majority of the literature in the field of systems for Ambient Intelligence concentrates on realizing mechanisms to tackle the problem of how to gather user information, match or recognize user behavioral patterns, or predict user actions, requirements and needs [26] [27] [65] [33]. Obviously, these activities that revolve around users cannot be overlooked because Ambient Intelligence is, above all, a user-centered paradigm. However, this fact should not be taken as a reason to relegate the context role to just considering those aspects that are only relevant for the user. On the contrary, context should be more globally considered to encompass the perspective of the goals and intentions that the context itself should comply with. For example, an Ambient Assisted Living context has among its goals and intentions the supervision and maintenance of the conditions that assure a safe completion of the daily home activities of the person who lives in such a context. Whenever something, such as an energy cut-off, takes place, the context should be responsible for leading the person to a safe place where he/she could safely wait until the light can be restored. In this case, the context should carry out those actions intended to comply with the context goals and intentions, as in this scenario is to protect the person’s integrity.

In this regard, extending the user-centered view to encompass the context should be considered as a key challenge for Ambient Intelligence. However, the focus should not be on determining how to specify goals and desires, but rather at how to enable the context to supervise the satisfaction of such goals and desires and how to return the context to its desired state whenever a disturbing situation has taken place. Despite the fact that stating the system goals can be successfully achieved, by resorting for instance to Multi-Agent Systems, there is much room for improvement in providing flexible responses that address undesired situations.

Besides, an additional weakness can also be pointed out as a direct consequence of the multidisciplinary nature that characterizes Ambient Intelligence. It has already been stated that the most compelling approach to tackling this multidisciplinarity is by means of a comprehensive approach [25] entailing different technologies and approaches. However, this is a highly demanding task that has generally been simplified by reducing the set of challenges being addressed, ignoring or overlooking the rest of them. The state of the art revision is also intended to identify how this weakness is managed by the different approaches.

Apart from the aforementioned weaknesses, there are some other relevant aspects that are of interest from the perspective of the revision of the state of the art in systems for Ambient Intelligence. In this sense, the following section revises the most relevant works proposed to date with the intention in mind of highlighting features worth mentioning, but also intended to determine those essential aspects of Ambient Intelligence that have been overlooked. Moreover, the remaining sections focus on analyzing how three of the most relevant challenges have been addressed. The first of those challenges, addressed in section 2.3, concerns how device communication demands have been fulfilled by means of middleware architectures. Special emphasis has been given to the revision of research approaches rather than on commercial implementations. Section 2.4 revises the solutions proposed to date to tackle the problem of knowledge modeling. Section 2.5 starts by revising the different existing strategies to service composition, and it continues with a revision of the implementation decisions that the previously surveyed architectures have adopted with the purpose of supporting service composition. Finally, the last section is devoted to summarizing the interim conclusions arrived at from reviewing the state of the art.

2.2 Ambient Intelligence Systems

The inherent complexity of building system architectures for Ambient Intelligence, as mentioned above, is an effect of the multidisciplinary aspect of the challenges involved in such tasks. As a major consequence, such multidisciplinarity has repercussions on the scale of the efforts that are required for accomplishing the development of such systems. As can be noticed from the following lines, the magnitude of the approaches proposed to date clearly reveals the need for addressing this endeavor from a divide-and-conquer perspective.

The different fields of knowledge from which the task of building solutions for Ambient Intelligence is being addressed, range from Artificial Intelligence to Robotics, Communication Networks, or Electronic Engineering among others. Some of the contributions made to these fields of knowledge can be successfully extrapolated to Ambient Intelligence, reporting great benefits in addressing some of the unresolved issues of Ambient Intelligence. However, this section is not concerned about describing such contributions from an implementation perspective, but rather it focuses on outlining the most relevant aspects of such contributions and how they helped bridge the differences with the envisioned scenarios for Ambient Intelligence.

At this point, it is worth mentioning the fact that research efforts in the Ambient Intelligence field are receiving important support from the European Union, as is evidenced by the significant number of projects that are directly funded.

Before starting the description and analysis of the surveyed works, it is necessary to state the terms that determine how such activity has been carried out. The following projects have been analyzed from the point of view of the shortages or weaknesses they are intended to solve, the approaches and solutions they propose and the details they have also overlooked. In the following lines, these projects and the architectures and approaches they propose are assessed from the perspective of two of the major concerns that this thesis is trying to cope with: a) how to deal with novel or unexpected situations; and b) how to implement the ad-hoc system behaviors that are supposed to lead the environment to the desired state. Nevertheless, these challenges have been overlooked for the majority of the approaches for Ambient Intelligence and, traditionally, attention in this field has been addressed to different concerns, such as to those related to human-centered computing, network management issues, service discovery and management, or Quality of Service, among the most relevant.

It is a general trend, among the following surveyed work, that despite successfully coping with some of the aforementioned common concerns, they tend to make simple assumptions about the set of situations that they are considering, therefore ignoring the possibility of unpredictable situations to take place. In addition to this, the spectrum of possible system responses implemented by these approaches is fixed and static, rather than being subject to the requirements and peculiarities of the specific ongoing situation. Despite having overlooked the management of unpredicted situations or the implementation of flexible responses, the surveyed projects have made breakthrough contributions to the field of Ambient Intelligence, and they are quite definitely the groundings upon which this and future works have to be built upon.

To start with, the emblematic MIT project called **Oxygen**[1] was one of the main and most successful approaches to the problem of pervasiveness. This project was mainly devoted to simplifying computer interaction, in such a way that efforts were addressed to achieve an interaction paradigm as "*pervasive as the air*". The relevance of this project is grounded in the fact that it was one of the first serious attempts to achieve a pervasive environment, based on the combination of the technologies that were populating the environment. Moreover, this project is mainly motivated by the desire to devise an interaction style that resembles the one that humans naturally use. In this sense, the intention is that rather than having to adapt human interaction patterns to those offered by machines, communication routines adapted to humans are provided. Basically, the Oxygen project can be considered as one of the firsts project aimed at placing humans at the core of computing.

The main focus of this project was to determine how to overcome the technical challenges that were involved in the advocated vision. The proposed approach consists in considering the environment as a combination of eight different technologies, each of which is specifically designed to address certain user needs. The different sets of technologies are arranged in such a way that interactions are offered in an easy-to-use and adaptable way, exploiting the benefits derived from the different technology cooperation as well as from the enhanced access to information based on customized information about user preferences.

Despite the fact that this project was not originally conceived to address Ambient Intelligence, it is worth noticing that the contributions it made have been foundational for the majority of the projects proposed for Ambient Intelligence.

The **Computers in the Human Interaction Loop (CHIL)** [147] project is proposed by a consortium of 15 research laboratories. The focus of this project is on how humans interact with computers, and how to turn this interaction into a more natural mechanism of communication in which people's attention is not drawn to the artifact, but on the contrary, it is the artifact's responsibility to focus on humans and their interactions.

Improving the interaction loop, in such a way that it comes to resemble human to human interac-

---

[1] `http://oxygen.csail.mit.edu/Overview.html`

tion, is addressed by coping with a set of challenges. One of these challenges consists in recognizing human activities. In order to overcome the disadvantages of a traditional reactive interaction, in which computers only respond to direct user requests, it is necessary to replicate the human assistant role, in favor of a more proactive mechanism. The idea behind the *assistant* simile is to devise a system capable of foreseeing user requests and to respond to them by offering users those services that might comply with their requirements. This challenge unavoidably entails the task of gathering data from an inherently dynamic and open world, which makes more difficult the interpretation of such data. No wonder the main hurdle is in understanding the meaning of sensor data as a whole rather than as isolated data.

An additional challenge addressed by this project is the lack of standards that characterize the current pervasive contexts. At different levels, the lack of standards affects the communication between different devices or the services they provide, or how to discover and integrate new devices appearing in the context, or how to combine information that comes from different sources. It cannot be ignored that the main concern of Ambient Intelligence is to enact a context in which devices are merged with the background. In this context access to information is provided in a pervasive and seamless manner. In order to address such challenge, it is an essential requirement to provide interoperability support among services and devices from different vendors. This challenge has been tackled by building a common and defined software infrastructure.

The project ended up in 2007, with successful results in a wide range of fields, such as those involved in perceptual technologies (person tracking, speech recognition, or emotion recognition) and services. However, the conclusions drawn after the end of the project lead to the need for further efforts in several directions. For example, despite claiming to use real data, these were only retrieved through supervising a meeting room. It is necessary to open up the supervised areas to additional scenarios and check whether the implemented mechanisms are capable of dealing with the situations taking place. Moreover, sensor data could have been of a greater help if they had been managed in coordination, rather than being considered in isolation.

The **Cognitive Robot Companion (COGNIRON)** [24] Project[2] is an initiative funded by the European Union, mainly intended to design and build a robot that, endowed with raw capabilities and the ability to learn and understand from interacting with humans, it is capable of imitating human behavior.

This project involves several multidisciplinary fields, such as human activity recognition, spatial awareness, or social skills, among some, achieving a breakthrough outcome in some of these fields of knowledge. It is worth mentioning the contributions made to the field of detection and understanding of human activity based on body tracking. Additionally, the context-sensitive path planning implemented by this project enables the robot to perform more intelligent planning, based among some, on the semantic knowledge it holds about specific cultural issues. The high level knowledge held by the robot is what enables it to perform human-like reasoning about plans.

Due to the fact that the COGNIRON project was mainly about Robotics, it can be argued that this should not be listed along with those that are specifically addressed to Ambient Intelligence. However, the conclusions and contributions of this project are synergistic to the Ambient Intelligence field, especially those involving learning and understanding capabilities, context-awareness, or decision making.

This project is also mentioned here because, despite being encompassed in a different field of knowledge from Ambient Intelligence, it does tackle the problem of how to address unexpected situa-

---

[2] `http://www.cogniron.org/final/Home.php`

tions. The non-determinism implicit in communicative activities requires a flexible system to support the human-robot collaboration. In order to achieve such flexibility, this project proposes an approach based on the characterization of the users behavior, from which *models of use* [23] can be derived that somehow minimize the unpredicted situations. So, therefore, they are not really tackling the problem of how to react to unpredicted situations, although they are providing a solution that minimizes the situations not previously considered.

Despite the fact that the characterization of user behavior proposed in [51] is specifically aimed at modeling the human-robot collaboration, it could be easily extrapolated to the Ambient Intelligence field, because the only difference lies in the fact that the COGNIRON robot adopts the role that *the environment* would perform in an Ambient Intelligence context. Those responses expected from the robot are also expected from an Ambient Intelligence context to come from the intelligent environment.

The third of the surveyed systems is the **Context AwareVision using Image-based Active Recognition (CAVIAR)** project[3]. This project is mainly devoted to extracting semantically enriched information from the analysis of image sequences. The abstract context model proposed for describing intelligent environments is especially interesting, initially in a rough way, that can later on evolve into more accurate descriptions of the environment, based on the interaction feedback. The human activity recognition task is grounded in a *situation network*, in which each situation is the result of a process federation, in which each process matches a human action.

This project is especially interesting for the approach it implements so as to recognize human activities, based on symbolic representation of video contents. Despite being limited to the information captured from the video analysis activities, the CAVIAR project shares some motivations with this thesis, as it is the fact that they are both intended to semantically enrich data extracted from the context in order to recognize the long-term activities that are taking place. The essential role that events play in understanding activities and human behaviors is also reflected in the approach that has been adopted in order to model the context knowledge, the Event Calculus logic [73]. This approach is mainly intended to model and reason about events and their effects. The work in [7] proposes the set of formal predicates that compose the Event Calculus dialect used for the CAVIAR project. However, from a common-sense point of view, these predicates are correct although incomplete. Some formal predicates are also required so as to model aspects such as the common-sense law of inertia, that states that things tend to remain the same unless externally affected. In this sense, the Event Calculus dialect designed for CAVIAR could not model the knowledge involved in the fact that the water level contained in a recipient is constantly increasing until it reaches the top of the recipient. In that moment, the water level stops increasing and the water spills over onto the floor. The predicate set proposed by the CAVIAR project has not been concerned about these and some additional issues of common sense.

Overall, despite being positively evaluated this project needs to broaden its application scope in order to consider not only the information extracted from video analysis activities but also from additional sources of information like sensors, domain knowledge, and common-sense knowledge. Overlooking any of these sources, as common-sense knowledge is ignored in the CAVIAR project, leads the system to a limited success whenever unexpected situations or events come into play.

The **Ambient Intelligence for the Networked Home Environment (AMIGO)** project[4] is an additional European effort, this time addressed to overcome the obstacles that are preventing *home*

---

[3] `http://homepages.inf.ed.ac.uk/rbf/CAVIAR/caviar.htm`

[4] `http://www.hitech-projects.com/euprojects/amigo/`

networking from being widely spread. In this endeavor, this project poses the focus at the middleware framework. Efforts are therefore addressed to the development of an open and interoperable middleware framework capable of supporting the set of services that ease and enhance the home activities involved in everyday life. This new home paradigm involves aspects that range from home care and safety to entertainment, in which technology interactions have been minimized and simplified so as to create the sensation that the environment is working for the user's needs rather than the opposite, in clear consonance with the postulates of the Ambient Intelligence paradigm.

Despite the field of application targeted by this project, the obtained results can be easily extrapolated to other fields. Of particular interest to this literature revision is the proposed architectural approach, which on the basis of a Service-Oriented paradigm divides the deployed platform into three main modules. These modules deal with the programming and deploying requirements, the middleware that facilitates the networking needs, and the final application and services deployed to users. Additionally, the design and implementation details upon which the proposed architecture is supported have been thoroughly documented in the multiple reports published in the project site. From these documents important conclusions can be drawn regarding the main project's contributions and weaknesses.

AMIGO is fully aware of the importance that interoperability issues have on the success of the proposed solution. Due to the interoperability problems that underlie the home devices, the AMIGO project is devoted to provide a unified and open access standard that enables home devices interoperability. In this regard, a set of translation mechanisms have been deployed in order to parse messages from an incoming protocol to the targeted one. Additionally, the interoperability support entitles the middleware framework for service discovery and interaction. Besides supporting interoperable services, this project is also devoted to providing intelligent services. However, intelligence issues are only considered with regard to how services are provided to the user, in a user-centered manner.

The adoption of a user-centered approach implies that the context notion is relegated to work for the user-centered view. On the contrary, the context role should occupy the same place as the user concept does, in such a way that both roles interact in determining the demands that are to be fulfilled by the intelligent services. Necessarily, not only users need to be considered in these scenarios but also the context. For example, it is a user requirement to maintain the home comfort, as stated in one of the scenarios considered by the AMIGO project. The detection of a fire alarm should be considered as a disturbance of the home comfort, even though users are not present at that moment. The system should be instructed to adopt a solution that minimizes the impact of a possible fire. However, the user-centered approach adopted by AMIGO overlooks this sort of situations.

Additionally, the context role has also been overlooked in the modeling task required for service composition purposes. However, this aspect will be lately analyzed in section 2.4, as part of the revision of the semantic model proposed to date in the field of Ambient Intelligence. It is not that the context notion has not been considered at all, rather, the problem is how it has been considered. The AMIGO project understands the context concept only from the point of view of the devices that are deployed in it and the services they provide. On the contrary, one of the working hypothesis of this thesis is that context has to be considered not only regarding the devices and services deployed in it, but also from the point of view of all the source of changes that might be relevant for the monitored context. This has been the working hypothesis of this thesis and one of the main points of disagreement with the semantic model proposed by the AMIGO project.

Additionally, this thesis shares with the AMIGO project the consideration of service composition as the most plausible approach to provide flexible architecture to Ambient Intelligence. However, there are some important differences that should be noted, like the fact that services in AMIGO are modeled as OWL-S semantic Web services [86]. In this sense, network and protocol issues should be abstracted from the service composition endeavor, and despite the interface implemented by services, all of them should be considered for composition. Moreover, an additional difference is grounded in how the composition requests are being specified. For the AMIGO project, it is the responsibility of the end users whereas, as stated in the introductory chapter of this thesis, the end purpose of providing service composition is to leverage the autonomy that should characterize systems for Ambient Intelligence. Nevertheless, the service composition aspect of the AMIGO project will be analyzed later on in section 2.5.

The **Ambient Intelligence for Mobile Communications through Wireless Sensor Networks (e-SENSE)** [109] project[5] is devoted to developing a synergy platform in which the different context technologies set in common the information they hold about the surrounding environment. The project is mainly founded on the information that can be retrieved from wireless sensor networks, which in turn is made available to the applications and services that are part of the Ambient Intelligence system. The architecture proposed under the umbrella of this project rests upon a middleware framework, which assumes the responsibility for managing and abstracting the aspects regarding communications, service discovery, or data processing, among some of the most relevant.

The main motivation behind this work is the important role that context information plays for Ambient Intelligence applications. It is therefore essential to standardize the task of collecting context information through the use of a wireless sensor communications system. This project proposes some sort of specific toolbox in order to tackle the difficulties due to the heterogeneity and dynamism that characterize these environments.

Despite the fact that the main contributions of the e-SENSE project have been addressed to the communication field of knowledge, specially to that of Wireless Sensor Networks (WSN), the work carried out in the field of activity recognition is also notable. Regarding the latter, the followed approach is based on the statistical characterization of the information retrieved from body sensors and its association to a fixed set of activities. This approach implies that an off-line analysis is required before an activity can be recognized. Furthermore, only those activities that have been previously considered can be recognized. The drawbacks and inconveniences of such an approach contrast with the reasonably good results that they have been claimed to achieve [83].

Especially interesting is the *activity classification model* proposed in [5] as part of the distributed architecture, devoted to recognize user activities out of event sequences. The proposed model counts on an alphabet of atomic activities that can be composed into composite activities. This approach is quite similar to that of service composition although inversely considered. Instead of using composite services in order to provide more complex services, the e-SENSE project resorts to the composition simile so as to identify the context activities.

Finally, it is also worth noticing the efforts addressed towards the measure of affective states. In this sense, this project is not only constrained to the users, services, and devices that are populating the context, but they also value the importance that emotions have on Ambient Intelligence, specially when human-machine interactions are involved. The work in [61] describes the experiment foundations and the obtained results. Despite its primitive stage, the results shown pose an appealing field for enhancing human activity recognition. It is also quite promising the envisaged benefits of an approach that considers human emotions as a constituent part of the context information that should be handled by the Ambient Intelligence system.

---

[5] `http://www.ist-esense.org/index.php?id=48`

The **MERL's Ambient Intelligence for Better Buildings**[6] project is devoted to the development of a platform upon which low-cost sensors are deployed. The idea behind this project is to employ the information retrieved from those sensors in order to improve the efficiency, productivity, and safety conditions of the buildings in which they are deployed. One of the main contributions of this project has been the *massive dataset* obtained from the sensor activations data collected over one year. These data can be analyzed seeking correlations that can lead to conclusions about behavioral patterns. For example, the work in [149] analyzes the existing correlation between soda consumption and the effect it has on physical activity by analyzing the sensor data gathered. This project has also been concerned with the ethical implications of Ambient Intelligence with regards to the use of images and sound recordings [117]. One of the main benefits of working upon the information retrieved from sensors is that intimacy is preserved in contrast to the information retrieved from images and sound.

Aside from the relevance of having gathered such a large collection of real data that provides an excellent dataset that can be used to evaluate correlation approaches [153], it is also worth mentioning the work in [154] that presents an analytical approach to social behavior, in an office space context. The analysis of the data gathered from the sensors deployed in an office context are intended to identify the time periods in which people tend to behave in an unusual way.

The **PERSONA Service Platform for Ambient Assisted Living Spaces** [141] project[7] aims at combining the benefits of the Ambient Intelligence tenets with those of Assisted Living, with the intention of developing a technological platform capable of supporting elderly or senior people in their everyday life. This project is supported in the contributions made by previous projects such as the aforementioned AMIGO project. One of the main challenges faced by this project is the construction of a self-organized middleware platform, capable of discovering and providing services in an ad-hoc manner. Interoperability among different technologies is addressed by means of an ontological approach. To this end, context knowledge is modeled as a context ontology, in which adaptability and composition rules have been described.

This project is mainly intended to perform activity monitoring and recognition. In this sense, the proposed approach is based on a modular solution to supervise, monitor, and interpret sensor data that are published to a communication channel known in this project as the *context bus*. In order to do so, it is necessary to transform sensor data into meaningful information about the activity that it is being performed. The adopted approach is based on the analysis of video sequences that provide a statistical inference of the activity being carried out according to the human posture adopted. Since the supervised contexts are constrained to domestic environments, postures have been classified according to the typical activities that can take place in such environment.

Rather than being yet another approach for context-awareness, this project points to the lack of appropriate means for context modeling as the main hurdle that justifies the lack of a breakthrough solution to the endeavor. In this sense, the PERSONA framework [44] is essentially grounded in a middleware architecture that considers *context* as the key element of the overall system. This approach supposes the continuation of the aforementioned AMIGO project, and therefore, they both share the same strengths and weaknesses stated previously. Additionally, among the new contributions made by this project, it is worth mentioning the proposal of a general purpose context reasoner, called *Situation Reasoner*, aimed at inferring new information out of the knowledge they hold about the environment and the current data describing an ongoing situation. However, the main drawback of this functionality is due to the fact that such inferences are accomplished through the use of fixed

---

[6] `http://www.merl.com/projects/ulrs/`

[7] `http://www.aal-persona.org/`

SPARQL queries. The scalability problems associated to RDF and SPARQL makes unfeasible to deal with a large-scale knowledge bases. Moreover, it seems quite inflexible since only foreseen queries can be issued.

Additionally, the service composition functionality has not been extended from the previous AMIGO version. On the contrary, composite services are still considered as a static combination of the existing services. It means that only those composite services that have been previously considered can therefore be offered to the end user. In this sense, the AMIGO project only considers composition through service specifications made by the end-user. The PERSONA project has extended that version so as to provide a certain set of composite specifications. In this sense, the service composition functionality provided by the PERSONA framework has much room for improvement so as to get to automatically service composition.

Also promoted from the European Union, the project **ALADIN - a Magic Lamp for the Elderly?**[8] [85] addresses the very specific concerns of Ambient Lighting Assistance. This project resorts to the simile of the Aladdin genie, and it expects the environment to respond to wishes, more specifically, the wishes concerning the lighting conditions. This project has been motivated by the certainty that light conditions have an important impact on human life. For that reason this project is also intended to assess the impact of light conditions in people's health and well-being. Moreover, this project is also intended to provide the means that automatically adapt the lighting conditions to match the most appropriate conditions. The main audience of this project is the elderly community. They can benefit from the general sense of well-being derived from providing appropriate lighting. It can also contribute to improving their sleep quality and improve and facilitate their autonomous living.

Despite being concerned with a specific piece of the overall picture of Ambient Intelligence, this project is interesting because of its market-orientation, and the great benefits of a basic and simple task such as supervising light conditions. The work in [84] provides a thorough analysis of the lightening conditions that surround people in their home environment. This analysis has also considered the different disabilities and aptitudes of the people in the context analyzed. The results obtained lead to the conclusion that no important achievements can be gained from simple approaches. However, it should not be forgotten that it also implies a hard task of characterization and modeling in order to determine how lightening conditions impact human well-being in both physical and mental states.

Currently, the project is still in progress, but important conclusions have been drawn so far. This project is concerned with the fact that social relationships cannot be replaced by pervasive interactions with the context. On the contrary, they provide a more interactive approach based on the use of human-like avatars [99], which resemble a person-to-person interaction, with which older people feel more comfortable.

The **SWAMI -Safeguards in a World of AMbient Intelligence-** [155] Project[9] is mainly concerned about privacy and security aspects of the Ambient Intelligence environments, what they have referred to as the dark side of Ambient Intelligence. It was previously mentioned in the ALADIN project that addressing lightening conditions contrast with those other surveyed projects that were considering Ambient Intelligence challenges as a whole. In this sense, the SWAMI project also adopts the ALADIN approach, and it solely focuses on one of the specific challenges of Ambient Intelligence, the security concerns.

The SWAMI project starts by describing four *dark* scenarios in which potentially harmful situ-

---

[8] `http://www.ambient-lighting.eu/`

[9] `http://is.jrc.ec.europa.eu/pages/TFS/SWAMI.html`

ations come into play. These scenarios have been used to identify the risks and vulnerabilities that should be considered when developing systems for Ambient Intelligence. Privacy, identity, and security are three major challenges that arise in pervasive contexts, such as those that characterize Ambient Intelligence, in which interconnected devices handle user information, sometimes, susceptible of potential risks. The approach advocated by SWAMI is to consider security as a constituent part of the development process, rather than being an add-on module that comes after having built the Ambient Intelligence framework. Additionally, privacy issues should not be ignored or overlooked, which means that it needs to be specifically protected. One of the main conclusions achieved by the SWAMI project was that the majority of the approaches presented to date violate the privacy boundaries. In this regard, new approaches to Ambient Intelligence should not only consider security issues as a constituent part, but it is an unavoidable claim that they have to make special emphasis on protecting human privacy issues.

The major contribution of this project is therefore the formalization and characterization of how security and privacy issues can be addressed by any attempt to build an Ambient Intelligence framework. Additionally, the ultimate goal of this project is to come up with a set of research and policy options that dictate the step sequences that should drive the achievement of security and privacy-aware system.

Finally, this thesis is not only framed in the European Union but also, in the Spanish research context. In this sense, it is worth mentioning some of the initiatives that are being addressed to provide solutions to Ambient Intelligence systems. In this regard, one of the currently ongoing projects, funded by the Spanish government, is being undertaken by a consortium of technological companies and universities, under the **mIO!** project[10]. This project is mainly devoted to achieving Smart Spaces through the use of mobile phones. To this end, the mIO! project is addressing efforts at designing and providing the appropriate technology capable of supporting ubiquitous services in Ambient Intelligence. This initiative pays a special attention to the key role that *context* performs in determining and characterizing the situation that surrounds end users. For that reason, significant efforts have been addressed to develop a context management infrastructure, adopting to that end a middleware approach.

Apart from the middleware framework, the role played by the proposed context modeling approach is also relevant. The work in [145] presents an ontology network especially devoted to modeling the user's contextual knowledge. Despite the fact that the semantic model that they propose in [108] is intended to support the modeling task undertaken in Ambient Intelligence, it has to be noticed that it is addressed from a different perspective than the one presented here. It is a user and device centered perspective that it is not concerned about the external factors that are affecting the context itself, and how it could evolve as a result of external events. Besides, an additional reason that motivates the proposal of a new semantic model rather than using the ontology proposed by the mIO! project is grounded in the need for conciseness and simplicity in the number of concepts and the relationships between them.

Additionally, it has to be highlighted that scalability and privacy concerns have been carefully considered. In order to represent and model the context information, this project has resorted to an ontological approach. In order to exploit the benefits of providing context-sensitive services, it is also essential to count on appropriate means to localize and to recommend services on the basis of the current context. It is still too soon to draw conclusions from the development of this project since it has just finished the first year of its four year duration.

---

[10] `http://www.cenitmio.es/`

2.3 Middlewares for Ambient Intelligence

The essential role played by communications in the Ambient Intelligence paradigms, makes it unavoidable to dedicate a specific section to revising the state of the art in middlewares for Ambient Intelligence. As for the previous section, the revision of the following projects has provided this thesis with a clear idea of the challenges that remain unsolved and those that have been successfully addressed by existing technologies.

The first of the revised approaches is the **HYDRA**[11] project [36]. This project echoes the difficulties of the middleware frameworks in supporting the communication and exchange of information of the different off-the-shelf devices. Different devices implement different protocols, architectures, or programming languages. This fact complicates the interaction among different vendor devices and poses the need for a mechanism that abstract low-level details enabling device interaction. This project is mainly motivated by the need to achieve a middleware framework that works as the abstraction layer that enables a transparent communication independently of the device. This project has therefore intended to develop a middleware solution capable of enabling seamless access to different devices, bearing in mind at the same time the need to comply with security concerns.

The most appealing aspect of the HYDRA middleware is the breakthrough statement they promote, as it is the fact that if device capabilities could be semantically described in such a way that an intelligent agent could understand and use them, the problem of device interoperability would have been bridged [64]. Since the proposed middleware implements a Service-Oriented Architecture (SOA), achieving interoperability among heterogeneous devices implies that services can be easily combined, in the sense of composite services, by means of semantic descriptions of the composition.

The notion of *semantic device* proposed by HYDRA is particularly interesting. It is used to describe the services that a particular application would desire a device to provide. In this sense, HYDRA makes a distinction between a physical device and a semantic device. At some point, this approach can be seen as a specific type of service composition, in a sense that the composite service is modeled in HYDRA as a semantic device. At the implementation level, programmers just need to deal with the semantic device, which will later on be statically mapped onto physical device services. In order to accomplish such static mapping, knowledge about such services and devices is required. By implementing such an approach, programmers are abstracted from dealing with all the activities that would be involved in the mapping tasks, such as service discovery, access to physical devices, and mapping onto the programmed instance. On the contrary, using the semantic device abstraction relieves the programmer from having to manually address the involved previous steps.

HYDRA has also taken care of easing the process of code generation for the semantic device abstractions. This has been achieved by adopting a Model-Driven Approach that, based on a descriptive model of the services provided by the semantic device, it automatically generates the code concerning the service call. Furthermore, it is also responsible for determining the physical device that is implementing the service, and prepare the data to be provided to the services.

The only detail that appears to be missing is the capability to automatically generate semantic services, rather than expecting programmers to specify them. In this sense, automating some aspects of the implemented model-driven approach, it could be possible to automate the generation of semantic devices. However, HYDRA does not consider this capability and the semantic device generation can only be carried out by programmers.

The **MORE**[12] project [45] is devoted to addressing the problem that underlies the way humans

---

[11] `http://www.hydramiddleware.eu/news.php`

[12] `ist-more.org`

interact with electronic devices, especially those embedded devices with wireless communication capabilities. This project is mainly intended to develop a middleware framework that conceals the complexity associated with the use of different technologies.

The MORE middleware framework implements a combination of both a service-oriented and a layered approach that is specially addressed to be deployed in embedded devices. Additionally, services are considered under the MORE middleware perspective as Web services, and therefore, all the services implemented on top of the MORE middleware have to be described using the WSDL.

The main contribution of this project is the proposal of the *group services* abstraction. Services are grouped according to the functionality they provide, and proposing common and unified access methods. Additionally, communication among group members is allowed in such a way that the information that can be provided by group services is more complete than that provided by a single service. The communication among groups is accomplished through any of the means of communication available to the groups. This way of communicating services can also be seen as a service composition mechanism.

Additionally, the management of these groups is performed by using policies of ternary elements of the form "*event, condition, action*". If an *event* takes places, the *condition* is evaluated, and if it is true, then the *action* is performed.

The fact that the MORE middleware is built upon the OSGi framework implies that services need to be developed according to the OSGi requirements. Despite claiming to support heterogeneous devices, only those services that have effectively implement the OSGi wrapper are considered by the MORE middleware. Additionally, the services deployed on top of the MORE middleware are Web services, whose semantics is limited to the short of descriptions that can be carried out using the WSDL. In any case, in order to make the most of the different capabilities of the devices deployed in such environments, it is necessary to semantically enrich such description so as they can be composed into more flexible and appropriate services. On the contrary, the MORE middleware only concerns about communicating those services that have been previously described as OSGi services.

The **MPOWER**[13] project [93] has been mainly motivated by the increasing number of elderly people suffering from different types of illness or disability. Inspired in the benefits with which new technologies can provide elderly people in their everyday life, this project is mainly devoted to develop a platform that eases the deployment of systems and services for the elderly and people with some sort of cognitive disability. The approach followed by this project consists of the combination of Agile Software Development and Model Driven Architectures, in order to support a rapid development and deployment of services.

It is worth mentioning the case scenario presented in [107] in which the focus is on trying to improve the living conditions of dementia patients so as to allow them to live in their homes as long as possible. The MPOWER system has been devised to be connected to a Care Center system, in which specialized staff is dedicated to monitoring and responding to the system alarms.

Regarding the implementation details, the MPOWER middleware adopts a Service-Oriented Architecture (SOA) approach, in which the system functions are provided as services. The main shortage of this complete approach is the lack of an intelligence module that could work to automate the monitoring and supervision tasks. In this sense, false negative alarms could be avoided and more importantly, information sent from the home environment could be reduced to a minimum, complying with privacy and security issues that seem to be overlooked in this approach. As for the previously mentioned approaches, services in MPOWER are also implemented as Web services, with the asso-

---

[13] `http://www.sintef.no/mpower`

ciated inconveniences.

The **MEDUSA** Project [31] proposes a middleware platform for supporting the Activity-Oriented Computing (AOC) paradigm. This approach is grounded in the enabling of application composition in smart spaces, as a mean to provide ad-hoc applications that support user activities once they have been recognized. However, despite the fact that under the AOC paradigm the focus is on leveraging autonomous systems that do not require user assistance, the truth is that involving users is still unavoidable in order to provide solutions in compliance with user preferences and desires. An additional limitation of the AOC systems is that service composition is only supported according to predefined activity descriptions and therefore, they cannot be composed in an *ad-hoc* manner. The solution proposed by MEDUSA is to enable service composition by entitling users to provide the description of the desired composite service or end application.

Despite being a flexible way of supporting service composition it is far from being automatic since a user description of the composition is required for it to take place. Additionally, the description of the composite service (or application as it is known under the MEDUSA project) is used by the *application composer* module as the route map that determines the services that should be part of the end application. To this end, an algorithm is used to perform the matching between the composition description and the service capabilities.

The interoperability issues that might arise when dealing with heterogeneous devices are addressed in MEDUSA by means of a *common network interface*. Additionally, the service interoperability is achieved through the use of a common service semantic model, in charge of establishing the mapping between the service concepts and the capabilities they imply.

The MEDUSA middleware could be easily enhanced with an intelligent layer that would be responsible for undertaking the automatic service composition task in response to user needs or requirements. On the contrary, only user specifications have been considered for composition. Moreover, as for the previous surveyed works, the fact that services are implemented as Web services also limits the integration and interoperability of those services that have not been previously wrapped into a Web service.

The previous paragraphs have surveyed some of the more recent approaches to the field of middleware architectures for Ambient Intelligence. However, there are some additional works that are worth mentioning as the approaches upon which most of the current works have been inspired. Despite being obsolete, the relevance of their contribution when they were proposed makes necessary a summary enumeration of the first contributions to the surveyed field.

The **Gaia**[14] [119] infrastructure envisages physical spaces as if they were computational systems. Under Gaia, these environments are known as *active spaces*, referring to any environment capable of sensing user actions through the sensor devices deployed in those environments. The Gaia approach has extrapolated the traditional theories of Operating Systems to that of Ubiquitous Computing. The proposed architecture is based on a set of functional blocks supported on the *Component Management Core*, which implements a middleware approach.

The **CORTEX**[15] [143] project is mainly devoted to develop systems capable of functioning independently of human supervision. These applications are built upon a set of sentient software objects, in such a way that these objects are capable of retrieving sensor data that works as object inputs. The main peculiarity of this approach is the communication mechanism it proposes. This mechanism is based on a publish/subscription paradigm of anonymous events, known as STEAM [92]. Addi-

---

[14] `http://gaia.cs.uiuc.edu/`

[15] `http://cortex.di.fc.ul.pt/index.htm`

tionally, the middleware architecture is arranged as a set of component frameworks in which each component is responsible for a specific functionality, such as service discovery, context, or resource management.

The **AURA** project[16] [67] concerns about the discontinuity problem caused by the device dynamism that characterizes a ubiquitous environment. The solution proposed by AURA is based on the notion of a personal *aura*. This personal aura is nothing else but a component in charge of managing resource availability, providing service continuity and supporting users in their high-level tasks.

The **SOCAM** architecture[55] is devoted to supporting the building and rapid prototyping of context-aware and mobile services for intelligent cars. The cornerstone of the proposed architectural approach is an ontology-based context model. This context model supports a wide range of tasks, that are basically intended to reason about the context and to support interoperability among context-aware systems. At the core of middleware architecture is the *context interpreter* module, in charge of reasoning about context and holding the context knowledge. Additionally, there are some other middleware modules that are in charge of providing context abstractions or service discovery and location. Finally, the context-aware mobile services are responsible for using previous modules in order to adapt the services to the current context.

2.4 Semantic Models for Ambient Intelligence

Since a semantic model is the main contribution of this work, particular attention should be paid to the work concerning this topic which has been performed to date. Despite the recent efforts of the W3C to provide a standardized and formal model of the environment, traditionally, there has existed a lack of consensus regarding the conceptual entities that should be part of the model. The Delivery Context Ontology [81] proposed by the W3C does not suffice to address the context-centered view advocated in this work. On the contrary, it is characterized for adopting a device-centered approach, in which the focus is on capturing and modeling the *context of use*. Aside from the context of use, additional issues should be considered in order to characterize and model the changes that make the context evolve from one situation to a different one. These aspects, however, have not been considered in the Delivery Context Ontology.

In this regard, the low level details with which the Detail Context Ontology has described the *environment* concept are also responsible for its rigidity and the impossibility to adapt such an ontology to different approaches, such as those focusing on users, user actions, or context events. This weakness has led to a situation in which each context-aware or Ambient Intelligence framework proposes their own specific model. The majority of the approaches tend to oversee the role played by the modeling task, and the justification as to why a model is composed of certain concepts rather than others tends to be overlooked. Among the concepts that should be modeled in a semantic model for Ambient Intelligence, solely the notion of context has been properly formalized by the work in [1]. Furthermore, based on the definition provided by Dey and Abowd regarding the context notion, the work in [118] goes a step beyond how the context notion should be handled. Three different levels of context are considered, partially ordered by sets. Whatever the cause may be, apart from the context concept and the Ambient Intelligence or context-awareness field, no relevant work concerning concepts such as actions and events haa been found which can be cited here.

Generally, the most common approach to context modeling is based on the use of a combination of OWL with some query language, such as SPARQL. The first shortage that can be detected in an ontology-based approach is due to the impossibility of attaching real meaning to the ontology

---

[16] `http://www.cs.cmu.edu/~aura/`

concepts. Concept meanings cannot be considered in isolation, and there is a great amount of knowledge involved in enhancing concepts with meaning. However, at some stage, it can be argued that the complexity involved in enhancing concepts with such a complex and large amount of knowledge can make the task of context modeling unfeasible. It is therefore necessary to find an equilibrium between the amount of knowledge required to provide a close semantic description of a concept, and the structure that allows such knowledge to be organized in a feasible way that can be used when reasoning and inferring about knowledge. Nevertheless, it should be remembered that ontology-based approaches have quickly achieved a high level of acceptance. The reason for this is mainly the fact that the knowledge modeled using an ontological approach can afterward be easily shared and reused.

The first of the semantic models surveyed here also implements the same ontology-based approach. The AMIGO project [114] proposes an architecture specifically devoted to managing context information, built upon a semantic model of the context information, which at the same time is supporting the information share among the different devices that populate the environment. The AMIGO semantic model understands context as being a physical context with different functional domains (i.e. PC, mobile, CE, and home automation). This project therefore proposes a complex structure of different ontologies, grouped in a modular manner [121]. The notion of action or event is ignored as it is the relationship of such concepts with the context devices.

Additionally, the OWL-S, RDF, and SPARQL have been the technologies used for the implementation of the semantic model and for querying it. As was mentioned above both technologies have scaling problems associated to them, since whenever the ontology is getting more complex, queries are taking exponentially increasing time in being evaluated. Moreover, SPARQL is basically a language for querying an ontology that does not support real inference mechanisms. In this sense, there is not much difference between querying a database and an ontology. Nevertheless, despite the drawbacks associated with the technologies employed, there are some strengths directly dependent on the semantic modeling of services proposed by the AMIGO project.

Regarding the service semantic modeling, the ontology-based service discovery and the dynamic service composition are two of the most challenging goals addressed by the AMIGO project.

The main contribution of the AMIGO semantic model is the tool they have developed to visually edit the context information ontology. The *VantagePoint* utility [70] facilitates the task of dealing with the ontology context information for those that have not been trained in knowledge engineering.

However, the semantic enhancement of services in AMIGO is basically intended to support semantic discovery and integration of Web services. However, the main drawback of this approach is grounded in the fact that the use of semantics has to be associated to Web services. There exists a dependency on the technology approach that needs to be implemented if services are to be semantically described.

Aside from the technological aspects and focusing on the proposed model itself, the main shortage of the AMIGO semantic model is due to the service-centered approach it adopts. Despite being labeled as semantic, the accomplished service descriptions are not grounded in a knowledge base that associates meaning to the concepts, but rather, this approach is claimed to be semantic just on the basis of the use of a common vocabulary. The role played by the proposed ontology is to standardize the concepts involved in service descriptions, however, such ontology should be supported, at a different level, on the meaning and knowledge that each concept and relationship of the ontology has associated.

However, ontology-based approaches are not the only strategies to context modeling, and the work in [139] and more recently the work in [11] present a survey of some the approaches employed for modeling purposes.

The idea behind the work proposed here is to combine, into a semantic model, the minimal set of concepts that are present and relevant to the different layers that compose an architecture for Ambient Intelligence (hardware layer, communication layer, and service layer) as the one advocated here. In this sense, this minimal core can afterward be enriched with further and more high-level details in the different layers. However, the core concepts composing the semantic model have to remain simple and common to all of them.

The work in [32] echoes the need for standardizing the semantic model proposed for context-aware or Ambient Intelligence frameworks, independently of the domain considered. This approach adopts a strategy based on answering questions such as: Who are the participants in the interaction?; Where does the interaction take place?; When does the interaction take place?; What does the interaction describe?; or how is context captured and accessed in the interaction? Once again, the proposed model fails to justify why these issues, rather than others, address these questions and should be reflected in the semantic model.

Some other projects resort to semantic models for different purposes: the work in [9] proposes a semantic model for services with the intention of supporting service discovery tasks in pervasive contexts; in [19] the semantic model, proposed as an OWL ontology, shares contextual information with agents; and finally, the authors of the work in [110] believe that the solution to supporting interoperability among devices populating the contexts lies in stating a common terminology. The semantic model is responsible for sharing such a common terminology.

Despite the existing differences concerning the concepts that should be present in the semantic model, when it comes to the modeling language, the use of OWL has become a common practice. The work in [48] describes a promising approach for dynamic service composition, in which the ontological model is represented by means of semantic graphs. The representation technology makes an important impact on the approach used to reason about the context. Refer to [11] for an extensive survey of context modeling and reasoning techniques.

2.5 Service Composition

With the emergence of Ambient Intelligence environments, dynamism, flexibility, and interoperability support are some of the main challenges to be addressed. As surveyed in the previous sections, to date, the main contributions to this endeavor fall into the field of service-oriented architectures and more specifically, implemented as Web service approaches. These circumstances enact the service composition as a suitable approach to tackle the need for capabilities that can be achieved by combining simple and existing services. This has been a major concern for the majority of middlewares for Ambient Intelligence. For that reason, this section is surveying the different approaches to service composition that can be found in the literature. The work in [15] provides a good categorization of the service composition mechanisms, however from the point of view of the *who*, *when*, and *how* the composition is specified. Nevertheless, the categorization proposed by the cited work is quite poor with regard to the different approaches that are supporting the composition task. For that reason, the following list surveys the approaches, proposed to date, to the service composition task.

- **Configuration files:** The first attempt to provide some sort of dynamism was based on the adaption concept. This proposal was intended to provide architectures with adaption capabilities by means of configuration files. These architectures counted on some features that could be customized in order to select the components loaded at the startup time. This approach allows a limited level of dynamism, far from what an adaptive middleware should be, since the main features are placed at the middleware kernel and cannot be changed. Therefore, rather than adaptive middlewares they should be considered customizable middlewares, supporting just a fixed group of cases.

- **Reflection:** As a second attempt, after the use of configuration files, reflectiveness appeared as the solution to add some dynamism support to middlewares. This proposal advocates for a middleware core with a minimal set of services installed in devices. By means of reflective mechanisms, applications can obtain from the middleware the context information, and use it to tune the behavior of the middleware.

- **Reflection and metadata:** The next stage in this evolution is based on the combination of reflection and metadata, aimed at developing adaptive and context-aware applications. This approach is mainly based on policies, that is, the use of a set of primitives aimed at describing how the context might change and how these changes are to be treated. Since conflicts among policies may arise, a solution based on a micro-economic approach was proposed in order to handle these conflicts.

- **Externalization:** Although reflective middleware services do support configurability, by supporting replacement and assembly of components in reaction to changes, the reality was that most of them assumed a basic backbone of fixed services. The externalization approach suggests a middleware architecture that explicitly externalizes the state, the logic, and the internal structure of middleware services, in such a way that the system can be updated, upgraded, or changes its configuration without requiring user intervention.

- **Policies:** This approach proposes the use of profiles, where the associations between services and policies applied to these services are described. Profiles are passed down to the middleware, and whenever a service is invoked, the middleware consults the profiles of the application that requests it. The profile determines which policy can be applied in the current context, depending on the state of the requested resource, thus relieving the application from performing these steps.

- **Web Services:** Among all these different approaches towards service composition, Web Services have been by far the most popular. This XML-based approach allows the specification of web services that can be dynamically loaded according to the requests. A service is specified by means of a service's abstract interface and the non-functional properties associated with the service. This approach provides a set of Service Repositories containing information about local and remote service repositories.

- **Ontologies:** Finally, it should be pointed out that nowadays, the use of ontologies is gaining great attention. Among all the ongoing proposals on this field, *domain ontologies* is one of the most relevant, intended to model the domain knowledge and provide semantics to service description. The capability to express semantic relations among services is quite useful in guiding the composition process.

Despite the fact that reflection and externalization are able to partially deal with interface heterogeneity, to consider a set of stated interfaces for each type of service helps improve the composition mechanism. In fact, the use of these types of stated interfaces is a key requirement for composing services under several approaches, such as those based on policies or web services.

Web service architectures provide a uniform treatment of services by means of WSDL, included in approaches such as IST-Amigo and WSAMI. However, it has to be remarked that the demand for resources under a middleware based on web services is too high, which in turn supposes the exclusion of limited resource nodes and therefore takes WSN out of a transparent integration.

Automatic service adaption is a key requirement when services use different interfaces and protocols. Generally, an adaptation process has to be carried out in order to inter-operate services belonging to different domains, for example, an UPnP and an Oxygen service. This requirement can be effectively addressed by means of proxies between services, or software wrappers for specific domains.

One of the least considered issues is dynamic service ranking. There exist several approaches intended to evaluate the service's suitability for a specific purpose. However, as far as the authors are aware, no attempts have been directly addressed to Ambient Intelligence middlewares. The service continuity requirement remains unconsidered by most of the approaches, basically, due to the fact that they make the simple assumption of having just one type of service in each scenario.

Despite the fact that the externalization approach has not been applied to Ambient Intelligence, it is interesting from the middleware point of view. This approach is adopted by the ExORB proposal, presented in [120], mainly aimed at mobile phones. This approach is aimed at constructing configurable, updateable and upgradeable middleware services that do not depend on user intervention whenever a change is required. Despite the fact that it offers support for dynamism, the reality is that it does not dynamically respond to changes in the environment, but it rather responds to developer's actions.

Attending to the most challenging features of service composition, Table 2.5 summarizes the contributions made by some of the most representative middlewares for Ambient Intelligence, as enumerated in the previous section.

The lack of a standard API (Application Programming Interface) for Ambient Intelligence services, similar to POSIX (Portable Operating System Interface for Unix) interfaces for operating systems, has delegated the service design task to developers. This is a major drawback since the lack of agreement supposes that similar services have different interfaces, making it more difficult to comply with interoperability demands and therefore, hindering the composition process.

Middleware	**Platform**	**Information Model**	**Definition of Services**	**Automatic Service Composition**
UPnP	HTTP/SOAP	XML	XML templates	Possible
OSGi	JVM/Java	No	Developer decision	Dynamic bundle invocation
AMIGO	OSGI/Web Services	Ontologies	Developer decision	Possible
HYDRA	Hydra middleware	Device Ontology	Semantic Web Services (SAWSDL)	Possible (Automatic Generation of SWS proxies)
MORE	MORE middleware (Publish/Subscribe)	MORE Integrated SOA Model	Web services	Possible (Service Chain Assembly)
MPOWER	MPOWER middleware	MDA/UML	Web services	Possible
MEDUSA	UbiSOAP and AmIi	non-specified	XML-based	Possible (Activity-oriented computing)
GAIA	Distributed Objects	No	Developer Decision	Possible
CORTEX	OpenCOM	Sentient Object Model	Task Model	Unspecified
AURA	Coda, Spectra, and Odyssey	Activity-based computing paradigm	XML-based	Possible (Task abstraction)
SOCAM	OSGi	Ontologies	OSGi compliant	Possibles (rules)

Table 2.1: Ambient Intelligence middleware features for service composition

Part II

Understanding

Chapter 3

Common Sense

Humans rarely learn "what" -we usually learn "which". In other words, we assimilate new information by finding similar things we already know about and recording the exceptions to that "analogy". This leads to amusing mistakes made by children: "*Will that Volkswagen grow up to be a big car?*"

– Marvin Minsky

Summary – *This chapter addresses one of the most characteristic features of humans, common sense, from a computational point of view.*

3.1 Introduction

Since its appearance, Artificial Intelligence has attempted to build systems with computational intelligence. This task has turned out to be a very difficult one, and despite the fact that computing systems have been improving their intelligence skills, the lack of common sense they suffer from is preventing them from becoming truly intelligent. In words of M. Minsky [97] "*some programs can beat people at chess. Others can diagnose heart attacks. Yet others can recognize pictures of faces, assemble cars in factories, or even pilot ships and planes. But no machine yet can make a bed, or read a book, or babysit.*" In his 1968 paper [89], McCarthy proposed an approach to build a program with the capability to solve problems in the form of an *advice taker*. In order to do so, McCarthy reckons that such an attempt should be founded on the knowledge of the logical consequences of anything that could be told, as well as the knowledge that precedes it. In this work, McCarthy postulates that "*a program has common sense if it automatically deduces from itself a sufficiently wide class of immediate consequences of anything it is told and what it already knows*".

Just before going any further into the problem of understanding and modeling human common sense, the meaning of such a concept should be clarified. First use of the notion of common sense is due to Aristotelian psychology, which conceived it in a comprehensive way, as the result of combining the five senses. However, common sense is nowadays understood in a different way, as the ability of human reason to use experience and previous knowledge in order to make deductions, establish connections and avoid contradictions [52]. In either case, the truth is that common sense has not been formally defined in either the field of Artificial Intelligence nor cognitive science, as stated in [37].

For D. Lenat [76], "*common sense is the sort of knowledge that an encyclopedia would assume the reader knew without being told (e.g., an object can't be in two places at once).*" M. Minsky [97] uses the term referring to the things that we expect other people to know, those things labeled as obvious. In this sense, sociologist have pointed out that the notion of common sense has a different meaning for each of us, since the things we consider obvious are subject to our social context, and therefore it is not unique.

Overlooking common sense when programming systems is the reason pointed out by M. Minsky as responsible for the deficiencies of current computational systems. In this sense, M. Minsky criticizes three aspects of current programs. The first issue is the lack of common-sense knowledge and the skills required to use such knowledge. For example, from the fact that a parcel is tied up with a string, there are many "obvious" facts that are straightaway associated with it. The string can be used to pull the parcel not to pull it, or the fact that if you push it too hard you might break it. These are some of the facts that despite being obvious for humans are unknown by computational systems, unless explicitly told.

The second issue refers to the lack of explicit goals, in the sense that systems are told "*what*" to do, instead of "*why*" to do it. This ignorance causes systems to fail in their task when something goes wrong, since no reasons have been provided. For example, "*people like to go indoors when it rains*", this fact tell us the "*what*", and the "*why*" is provided by the fact that "*people do not like to get wet*". Knowing that fact makes it possible to fulfill the ultimate goal of preventing people from getting wet, for example by using an umbrella, if going indoors is not a possibility.

The third issue is related to resourcefulness and how people use common sense to make analogies when some required knowledge is missing.

Due to the fact that the theory of common sense is one of the cornerstones of this thesis, special attention is paid to summarize its foundations. The second section revises the most important attempts proposed to formalize common-sense knowledge. The third one deals with the tenets or key issues of common-sense reasoning. For further information about the contents of these two sections please refer to [88][142][100]. The fourth section revises the common sense systems proposed to date with regard to the requirements demanded by systems for Ambient Intelligence supervision. The fifth section discusses how common sense can be endowed to Ambient Intelligence. The sixth section proposes a list of benchmarking problems for assessing the performance of a common-sense reasoning module deployed in an Ambient Intelligence environment. Finally, the last section sets out the interim conclusions drawn from the role that common sense can play in a system devised to supervise Ambient Intelligence environments.

3.2 Systems for common sense reasoning

Automating common-sense reasoning has been one of the primary concerns for researchers in the Artificial Intelligence field. According to E. K. Muller, who in [100] provides a brief history of common-sense reasoning, the first work in this field dates from 1956. The main contributions to this field come from authors such as A. Newell, who mainly worked on the cognitive aspect of Artificial Intelligence; M. Minsky, who has made enormous contributions in the domain of common-sense knowledge representation and reasoning [96] [97]; and finally D. Lenat, who undertook in 1984 the first real attempt to catalog common-sense knowledge, under the Cyc project [77].

Automating common-sense reasoning is a task that requires a sufficiently expressive language, a knowledge base to store such a large amount of knowledge, and a set of mechanisms capable of manipulating this knowledge, so as to infer new information. Regarding the knowledge base, Cyc[77], ConceptNet [82], Scone [42], and WordNet [94] are by far the most evolved and successful approaches found in the literature.

Cyc has formalized the largest body of fundamental human knowledge to date. Nowadays, Cyc Corp is addressing its research efforts to automate knowledge acquisition, either by interacting with people [152] or by making use of the already asserted knowledge, the natural language understanding, and knowledge published on the Internet [131]. In contrast to the proprietary system approach followed by Cyc, ConceptNet resorts to the general public for acquiring knowledge. ConceptNet adopts a semantic network structure similar to WordNet. Nevertheless, when compared, ConceptNet claims to hold more informal, defeasible, and practical knowledge. It can also be argued that WordNet should not be listed along with common-sense reasoning systems like Cyc and Scone, as it is just a large database of English lexicon.

Scone is an open-source knowledge based system written in Common Lisp. The main difference with respect to other approaches is found in the way search and inference are implemented. Scone adopts a marker-passing algorithm [40] devised to be run in the NETL machine [41]. Despite the fact that these marker-passing algorithms cannot be compared with general theorem-provers, they are indeed quicker, and most of the search and inference operations involved in common-sense reasoning are supported: inheritance of properties, roles, and relations in a multiple-inheritance type hierarchy; default reasoning with exceptions; detecting type violations; search based on set intersection; and maintaining multiple, overlapping world-views at one time in the same knowledge base.

The following subsections are devoted to the identification and analysis of the main features of the aforementioned approaches to common-sense reasoning. Basically, the focus is assessing the strengths and weaknesses of the different systems, in order to justify the election of Scone over the existing approaches. Three main aspects are considered for evaluation, the modeling and query language, the knowledge-base structure, and the reasoning engine.

3.2.1 CYC

The Cyc project, led by Doug Lenat, was envisioned as an approach for building a large scale knowledge-base system dedicated to holding and acquiring common-sense knowledge. The ultimate goal of this endeavor is to enable this system to autonomously acquire new knowledge imitating the learning process of children.

This project, rather than being grounded in complex theories of intelligence or human cognition, was intended to build a pragmatic solution to the problem of building a large scale knowledge-base system of common-sense knowledge. Cyc, on the other hand, is built upon two premises [78], as known: a) those aspects regarding human intelligence do not have any relevant impact on the Cyc project; and b) common-sense knowledge is the only requirement for intelligence. Aside from common-sense knowledge, human knowledge is also grounded in the rules and heuristics attached to human cognition and human behavior. In this sense, despite the pragmatic approach claimed by Cyc, some aspects of human intelligence have definitely been considered by the Cyc project.

The analysis of how human knowledge is organized in the Cyc knowledge-base gives rise to the importance that the *context* concept plays in this endeavor. Nevertheless, the context concept has not always been given the same importance as it now has in Cyc. On the contrary, three different stages can be identified in the role of such concept [37]: a) the context concept lacks of significance; b) Cyc adopts an approach based on *microtheories* so as to encompass the knowledge of a specific context; and c) context is considered in this stage as a twelve-dimension reality (absolute time, type of time, absolute place, ect.).

Time is modeled in Cyc as a two-dimensional space, one for the *truth* value and a different one for the *relevance*. This approach leads to situations that make the statement "*Bill Clinton is the president of the United States*" true even before Bill Clinton was born, on the basis that in 1900 this fact was irrelevant since it did not affect any aspect of that moment. Consequently, the context concept is divided into four subcontexts, each of which contain assertions that are either true, true and relevant, true throughout a time interval, or true just during a specific period of time. These four subcontexts provides the basis for a more realistic modeling of changes.

3.2.2 OpenMind

Motivated by the same problem of providing computers with the ability to reason in the same way humans do, the project focuses at gathering and providing computers with common-sense knowledge by making the most of the Internet era and Internet users' collaboration.

This approach is grounded in the fact that if people could contribute with small pieces of knowledge about how the world works, then the problem of gathering such a vast number of assertions becomes a feasible problem [133]. However, due to the fact that the majority of people lack of the required skills to assert such knowledge, an appropriate way has to be devised so as to ease the process. The response to that requirement is the site at www.openmind.org/commonsense, which has turned the assertion of common-sense knowledge into an easy and amusing process.

This approach is said to be inspired by Cyc, and for that reason, they both share some similarities regardless of the different approaches followed for gathering knowledge. An additional difference between Cyc and Open Mind is in how they express common-sense knowledge. Cyc uses its own representation language, CycL, whereas Open Mind uses a reduced version of natural language in plain English.

In order to cater for the comprehensive endeavor undertaken by the Open Mind project, some additional systems are required. ConceptNet[82] is the semantic network derived from the transformation of common-sense knowledge stored at the knowledge-base. In this semantic network, graph edges represent common-sense relationships between two concepts. Additionally, each assertion is also graded with a score that depends on the number of votes or similar contributions made about it. Aside from ConceptNet, the AnalogySpace [138] strategy is devoted to overcoming the drawbacks associated with large knowledge-base systems when it comes to reasoning with the knowledge it holds.

3.2.3 Scone

The Scone project, led by Scott E. Fahlman at the Carnegie Mellon University, represents an open-source knowledge-based approach in which the focus is not on collecting common-sense knowledge but rather at providing the means for supporting common-sense reasoning mechanisms. The Scone system therefore pays special attention to providing an expressive, easy to use, scalable and efficient approach for accomplishing search and inference operations.

The main difference between this and other approaches lies in the way in which search and inference are implemented. Scone adopts a marker-passing algorithm [40] devised to be run in the NETL machine [41].

One of the main objectives Scone was conceived for was to emulate humans' ability to store and retrieve large pieces of knowledge, along with matching and adjusting existing knowledge to similar situations. To this end, the multiple-context mechanism implements an effective way to tackle this objective. The multiple-context mechanism also provides an efficient solution by which to tackle a classical problem of Artificial Intelligence, the frame problem.

The great potential of the multiple-context mechanism used by Scone can be better stated by using the example described in [40]. Since "Harry Potter World" is quite similar to the real world, a new context, "HPW", could be created as an instance of the real world[1]. Nevertheless, there are differences

---

[1] In Scone terminology, "general" is the context node that holds knowledge about the real world, and "HPW" would be an individual node, connected by an is-a link to the "general" node.

between these two contexts, such as the fact that in the “HPW” context a broom is a vehicle. This fact can be easily stated in the “HPW” without affecting real world knowledge, in the same way that knowledge of the real world could be cancelled so as to not be considered in the “HPW” context. The way in which Scone handles multiple contexts so as to avoid incongruence problems is by activating one context at a time. By doing this, only the knowledge contained in the active context is considered for the reasoning and inference task.

Unless otherwise stated, the knowledge described in a parent context is inherited by the child context. The context itself is also a node and, like the other nodes, it stores a set of marker-bits. One of these marker-bits is the context-marker. This bit, when enabled, determines the activation of all the nodes and links that are connected to the active context.

3.3 Key issues of common sense

Building intelligent systems is a task that requires an expressive language, a knowledge base capable of managing such knowledge in an efficient way, and a reasoning engine that uses such knowledge to deduce or infer new information. Aside from those requirements, there are some characteristic features [100] of common sense that need to be addressed by any attempt intended to endow systems with intelligence. The following subsections analyze all those issues or requirements from the perspetive of what needs to be satisfied so that common-sense reasoning capabilities can be enacted.

3.3.1 Representation

The different purposes to which the represented knowledge can be dedicated to also determine the type of formal logic that better fits the intended end. Representing common-sense knowledge involves collecting the vast knowledge about how world works. This endeavor demands a sufficiently expressive language that does not fail to be computationally efficient when managing such a large amount of knowledge. Despite being very expressive, natural language is not a choice for several reasons, such as ambiguity, and the need for a mean to perform reasoning about such knowledge representation. On the contrary, formal languages overcome the aforementioned drawbacks by giving support to the reasoning mechanism or avoiding ambiguity by means of precise definitions.

Among existing formal languages, this section revises the main feature of those that can be used to model and reason about common-sense knowledge. Situation and event calculus are two of the logic-based languages summarized here. Aside from those logic-based approaches, CycL and NETL are also analyzed and compared.

Situation Calculus

In 1969 McCarthy and Hayes introduced in [88] a formalism for representing world changes known as situation calculus. Change can be modeled as transitions among situations. For example, dropping an object can be seen as the situation in which you are holding the object and the next situation in which the object has crashed against the floor.

Situation calculus is a first-order logic dialect, although it can also be enriched with second order features, which basically includes three special concepts: *situations*, *actions*, and *fluents*. *Situations* are used to refer to possible worlds histories that, rather than being extensively described, use facts to describe the situation. Since it is unfeasible to fully describe all the facts that are involved in a situation, starting from the provided facts, additional facts are deduced from them as well as future situations or situations that may arise. Therefore, not only plausible situations have to be considered, but also any hypothetical ones. *Actions* play the role of leading from one situation to a different one, in other words, actions are responsible for a dynamic world to change. *Fluents* are those functions that, subject to situations, denote the effects of actions. *Propositional fluents* are those functions whose application domain is *(true, false)*, whereas the *situational fluent* domain is that of the situation set itself. For example, the fluent $rain(x,s)$ asserts the fact that it is raining in place x in the situation s.

Since the dynamic world is described in terms of situations, actions, and fluents, reasoning in the situation calculus involves making decisions on the knowledge of the available actions that can be performed by an agent and the fluents that will describe the changes caused by those actions.

Reasoning about the changing world therefore involves holding both knowledge about initial true facts and knowledge about how actions result in world changes. Some actions also need to count on the satisfaction of certain *preconditions* so as for the action to take place. For example, in order to pick an object up, it is a precondition to approach the object. Letting predicate $Poss(a,s)$ be used to denote that action a can be performed in situation s, and resorting to a more elaborate example extracted from [12]:

$$Poss(pickup(r,x),s) \equiv \forall z\ \neg Holding(r,z,s) \wedge \neg Heavy(x) \wedge NextTo(r,x,s)$$

Actions are also characterized by the causing *effects* in the sense of fluent changes as a result of the performed action. For example, the effect of dropping a fragile object is that of breaking it [12]. Letting the situation term $do(pickup(r,x),s)$ denote the situation where agent r picks the object x up in the situation s:

$$Fragile(x) \subset Broken(x,do(drop(r,x),s))$$

Aside from the concepts of situation, action, and fluent, situation calculus also resorts to an additional set of special predicates and functions, such as the above mentioned $do(a,s)$ and $Poss(a,s)$ stating in the first case that situation s is the outcome of performing action a and in the second that action a is possible at situation s. Additionally, $Holds(p,s)$ states that fluent p is true in situation s.

Nevertheless, situation calculus is not constrained to the aforementioned predicates and functions and additional ones can be specified. For example, it can be useful to define a ternary predicate so as to assert causality relationships by means of the predicate $Caused(p,v,s)$ meaning that fluent p causes fluent v to be true at situation s [142].

Original Event Calculus

The event calculus is a first-order logic formalism, proposed by Kowalski in [73]. This way of formalizing common-sense knowledge takes special note of the role that time plays in representing the world dynamics.

Despite the fact that situation and event calculus have much in common, they differ in how time is handled in each of these formalisms. Situation calculus manages time in a tree-like fashion whereas event calculus considers time in a linear manner. As stated by Kowalski in [73] the difference between event and situation calculus is just conceptual, such that situation calculus represents global states whereas event calculus deals with local events that take time at specific time periods.

Apart from that distinction, both formalisms consider actions, known in event calculus as *events* and *fluents*, denoting those time-varying properties.

The event calculus is specifically targeted at representing event occurrences rather than describing the action sequence at which situation calculus is especially aimed.

The following table [2] has been extracted from [142] and summarizes the predicates and functions of the original event calculus proposed by Kowalski and Sergot[73].

---

[2] Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter, editors. Handbook of Knowledge Representation (Foundations of Artificial Intelligence). Elsevier Science, 2007. pp. 701

Basic event calculus

The event calculus language proposed by Kowalski and Sergot in [73] and already analyzed in the previous section only concerns *discrete change*. Situation calculus also works in that direction and represents world dynamics as an ordered sequence of "*snapshots*", known as situations.

Nevertheless, none of these formalisms provides the means to represent the continuous change experimented by the level of the water in a kitchen sink with the plug in. This shortage motivates Shanahan to postulate an extension of the event calculus that could handle continuous change [132]. Mueller uses the term *common-sense law of inertia*[100] to refer to that property of things that make them stay the same unless externally affected. However, some properties needs to be released from this common-sense law of inertia and permitted to vary. As in the example of the water level, it should be released from the common-sense law of inertia while it is increasing until it reaches the top of the sink. When that happens the water level is stabilized and it can be once again subject to the law of inertia.

CycL

So far this section has revised representation formalisms which correspond to first-order logic dialects. However, logic-based formalisms are not the only choice when it comes to representing knowledge and *frames and slots* and *object-oriented* are two additional approaches that can be used to this end.

CycL is the Cyc representation language [79] and therefore it has been specifically designated to represent common-sense knowledge. Originally devised as a frame-based language, currently its syntax consists of a first-order predicate calculus augmented version [104], and it is also based on Lisp [28].

Time-varying properties, referred in the situation and event calculus as fluents are represented in CycL in two different ways. One of these ways is by considering existing objects as a combination of *space-intelligence-time* events [79]. For example, a person is considered in the Cyc ontology as a unit, having his birth as the *startingTime* and his death the *endingTime*. In between those two points different slices can be pointed out. These phases are referred as *temporal subabstractions* of the given object. The example provided by Mueller in [100] uses temporal subabstractions to represent the occupation temporal-varying property of the person Nathan. If in 2007 Nathan was working as a scientist, on March 4, 2007:

Table 3.1: Predicates and functions of the original event calculus

Predicate/function	Meaning
$Holds(p)$	p holds
$Start(p,e)$	e starts p
$End(p,e)$	e ends p
$Initiates(e,f)$	e initiates f
$Terminates(e,f)$	e terminates f
$e_1 < e_2$ e_1 precedes e_2	
$Broken(e_1,f,e_2)$	f is broken between e_1 and e_2
$Incompatible(f_1,f_2)$	f_1 and f_2 are incompatible
$After(e,f)$	time period after e in which f holds
$Before(e,f)$	time period before e in which f holds

(SubAsbstrac Nathan Nathan2007)
(SubAsbstrac Nathan2007 Nathan20070304)
(occupationOf Nathan20070304 scientist)

Temporal-varying properties can also be represented in CycL by using the *TemporalThing* collection, in a way closely related to the aforementioned notion of fluents. *TimeInterval*, *TimePoint*, and *SomethingExisting* are specializations of the *TemporalThing* collection, and therefore a statement such that (holdsIn t f) asserts that formula f is true throughout "t". Mueller in [100] provides us with the following example:

(holdsIn
(DayFn 4 (monthFn March (YearFn 2007)))
(occupationOf Nathan scientist))

Regarding the notion of event, CycL resorts to the *Event* collection, as a specialization of the *Situation-Temporal* collection, to denote the event occurrence, adopting the Davidsonian approach. However, preconditions and effects of actions and events do not count on a standard procedure for their representation, but they are rather susceptible to the use of a variety of predicates [105]. Although OpenCyc[3] implements a Hierarchical Task Network HTN-like planner, and therefore it considers the representation of preconditions, context-sensitive effects, and compound events, this is done in a handcrafted and very rudimentary manner, obviating most of the features that characterized common-sense knowledge that will be discussed below.

3.3.2 Reasoning

Formally, the art of reasoning consists in combining the use of formal languages along with judgments and inference rules intended for theorem postulations. Basically there are two main types of reasoning, namely: deductive and inductive reasoning. The former allows conclusions to be drawn given a specific set of assumptions, in such a way that true assumptions leads to true conclusions. On the other hand, inductive reasoning goes from the specific to the general, making generalizations about observed facts. This sort of reasoning therefore does not require conclusions to be true given a set of facts or assumptions. Category-based induction, analogical reasoning, or mental models are, among some, more complex sort of reasoning. Aside from these types and on the basis of the event calculus, there are several reasoning types that can be performed, such as abduction, postduction, and model finding. Similarly to planning, abduction determines the course of actions that leads from an initial state to a target one.

Due to the fact that common sense is one of the main topics of this thesis, this section basically concerns about common-sense reasoning. In order to be automated or computationally performed, reasoning needs to count on symbolic languages so as to translate knowledge into computational representations that can be managed by the reasoning system. Talking about common-sense reasoning therefore refers to the conclusions that can be derived from common-sense assumptions. The ultimate goal of endowing intelligent systems with common sense consists in enacting the systems' capability to use the knowledge that describes current situations so as to produce new knowledge as a result.

The reviewing of the state of the art for common-sense reasoning leads us to conclude that there are several well-identified areas in which work is being done in isolation. In other words, common-sense reasoning is being individually addressed from the many different aspects that make up the

---

[3] `http://www.cyc.com/opencyc`

issue, namely, default or non-monotonic reasoning, temporal and spatial reasoning, actions, situations, and events reasoning, belief, desire and intentions, also known and mental model reasoning, and finally casual reasoning [101].

The complexity involved in common-sense knowledge prevents research efforts from being comprehensively addressed to support common-sense reasoning. According to the approaches discussed, the common-sense reasoning problem has been divided into isolated issues that have been individually addressed. See [101] for a thorough revision of the works proposed to date about the theory of common-sense reasoning. The following subsections accounts for some of the most representative challenges for common-sense reasoning.

3.3.3 Effects of Events

The notion of event, aside from being considered by Davidson as part of the common sense ontology, also plays an essential role in the domain knowledge of context supervision. Therefore, special attention is paid to modeling events and their effects.

The Davidsonian view of what events are and how they can be individuated is supported by the knowledge about what caused the event to take place and what the effects of the event's happening have been. Only by considering cause and effects can events be individuated in such a way that events x and y are considered equals when they have the same causes and effects [29]. Davidson later on proposed a new conception of individuation of events, based on spatiotemporal locations.

The importance that the notion of events and their effects has on the development of intelligent systems is basically founded on its close connection to action planning. Knowledge about how certain events can affect the world can be used by an action planner to devise the course of events whose effects will produce a desired state of the world.

The difficulty in modeling effects of events arises when additional considerations need to be taken into account, some of which cannot be definitely stated at the modeling stage and are therefore subject to the context.

Events cannot be considered in isolation but rather as a constituent part of the context in which they take place. The same sort of event might give rise to different effects if they are considered in different contexts. For example, given two contexts, one of which considers an object on the table while the other one depicts the same scenario but with a slippery object instead, the event of picking up the object will likely result in two different effects taking place. The effects of picking up an object result in the object being held by the person who picked it up. However, in the second scenario, if the person does not pay enough attention in picking up the object, it is likely that the object will slip away and it will be dropped. This sort of effect is known as *context-sensitive*, and in order to be determined, events involving context-sensitive effects need to be framed in a context.

Sometimes, context information is not enough to determine or unveil in advance how an on-going event will result after taking place. Those effects of events that cannot be determined beforehand are known as *non-deterministic effects*. For example, in the previously stated scenario where the slippery object was involved, ignoring if the person that picks it up is going to be careful enough to avoid dropping the object leads to uncertainty about the resultant effects.

Non-determinism is associated with unawareness or ignorance of a context property just before the event takes place. Note how the uncertainty associated with irrelevant properties of the context does not have an impact on foreseeing effects of events. Not knowing the color of the object to be picked up does not interfere with the effect when the pick it up action is performed. It is not the case of the other properties as it is the object's texture that might alter the consequences of the picking up event.

Moreover, events not only cannot be considered in isolation with regard to their context, but neither with regard to other events. Effects of events are also affected by the effects of concurrently happening events which can alter the expected results. For example, the expected effect of pushing a trolley is that of the trolley being moved forward, while the effect of pulling a trolley is that of moving it backwards. However, if we consider both events concurrently taking place, the resultant effect of pulling and pushing the trolley at the same time is that of the trolley remaining static. When several events occurring at the same time do not lead to the expected effects but rather nothing happens, then these events are said to produce *canceling effects*.

On the contrary to canceling effects, concurrent events might also cause *cumulative effects*. These sorta of events experience an increased effect as a result of several events taking place concurrently. For example, think of two concurrent events each of which consists in pushing a trolley. In this scenario, the concurrence of these two events results in an increased momentum being applied to the trolley, which will be translated into an increased speed of movement when pushing forward. The occurrence of side events needs to be considered prior to being able to determine the expected effects of events.

Besides the cumulative and canceling effects of concurrent events, it can also happen that the concurrence of several events does not affect each other or, on the contrary, cannot be possible. The later refers to the set of events that cannot concur attending to some other rules of nature. For example, a person cannot be in two different locations at the same time. Ignoring the fact that a person can be in the kitchen and in the house at the same time, as the kitchen is inside the house, this restriction rather refers to the impossibility of being in two different physical locations[4]. Considering that the kitchen is physically located in a different space than the living room, a person cannot be at the same time in the kitchen and in the living room; that person is either in the living room or in the kitchen, but not in both places at the same time.

Moreover, effects of events might not only be constrained to the objects that are directly involved in on-going events but, additionally, their effects might expand to third party objects. These effects are known as *indirect effects*. Think of the scenario depicted above for the on-going event of picking up an object and now consider the occurrence of a new event in which the person that holds the objects change his/her location. The event of moving from one place to another causes, as an indirect effect, the location change of the object being held by the person in movement.

Certain types of relationships that alter inner properties of objects are susceptible to being indirectly affected as a result of the connection being affected. The spatial location of an object is one of those inner properties that is affected when an object establishes a relationship of *being hold by* with a different object. In such situations, the spatial location of both objects is merged into one and whenever one of them changes its location the other one unavoidably comes along.

Events can be considered to take place at an instant time point or, on the contrary, to take place over a time interval. The latter are more likely to cause continuous change or delayed effects. The effects of those events that happen to take place all along a time interval are not expected to cause effects after the event's completion, or to cause effects while the event is taking place. For example, the event that turns on a faucet whose sink has the stopper in results in a water level increase until the point where the height of the water level reaches the top of the sink, after that moment, the water will be spilled. The increasing water level is a continuous change effect while the spilled water can be considered to be a delayed effect. The event of flowing water into a closed sink is considered to last while the liquid is flowing. These events that take place over a time interval are likely to produce

---

[4] The difference between being *at* a place and being *in* a location will be explained later on when dealing about space.

continuous or delayed effects.

3.3.4 Space

Dealing with space is a task that requires the space to be divided into three-dimensional pieces, which are necessarily limited by physical boundaries and are somehow, connected one to another. Those boundaries not only delimit spaces but also serve as the linking elements between the connected spaces. For example, one of the walls that bounds a room can also be the element that connects that room with a neighboring room, and therefore, if there is a doorway in it, it can serve as a way through which to get to the other room. Describing spaces involves both knowledge about what the boundaries of such a space are and also how it is connected to neighboring spaces.

It is also important to know how different entities can move among spaces and how boundaries influence such movement or transition. Notice how liquids move from one container to another by raising the liquid level above the height of its container while a person moves to a different room by simply crossing the doorway that separates them. The object's quality is to determine how boundaries are breached.

The main implication of containment is its influence on causality [57]. Spaces also prevent effects of events from affecting those entities that are not directly reached. However, boundaries are not equally effective in preventing all types of effects, and while a brick house prevents those inside from getting wet when it rains, a paper house does not. Therefore, space and boundaries might influence how effects of events spread but only when effects cannot go beyond the given boundaries.

However, will the water of the rain eventually get inside the house? Common sense has taught us that houses have breaches in those points where different types of boundaries are connected. For example, doors are likely to filter liquids inside through the gaps between the door and the doorway. The granularity of space might vary depending on the situations where it is considered. When dealing with big spaces such as a room, it is suitable to think of a perfect contact between surfaces. Nevertheless, it is not totally true if we consider it not only from a microscopic point of view, but also from that of flowing liquids leaking through the gaps between the door and the doorway.

Related to the notion of space, the words *place* and *position* seem to be used indistinctly in natural language. Nevertheless, the notion of position refers to an exact point, in the sense of point in a Cartesian plane, while places have the implicit hint of existing boundaries. Despite the fact that positions are also places, as stated in [57] "a position is a place you can be *at*" whereas "a place is a place you can be *in*".

3.3.5 Common-sense law of inertia, change and time

As has been already stated when describing the foundations of the Situation Calculus, actions and events are described as functions that given the current state result in a different one. Therefore, the description of actions and events is indivisibly associated to the descriptions of the changes they produce in the context where they take place.

Regarding the temporal aspects of changes, they can both happen instantly or last a finite time. The latter happens to cause changes throughout the time interval, producing therefore *continuous changes*. The change produced by the action of dropping an object is that of the same object decreasing its height position until hitting the ground. The continuous change of the height property can be modeled by means of the "*trajectory axioms*". These axioms describe how the affected variable, in our example the height, changes with time, as a function of its initial height and the forces acting, i.e. gravity.

Resorting, once again, to the Situation Calculus proposed by McCarthy, changing variables were represented by means of the notion of *fluent*. Combining dispatching actions or events with fluents, it can be described how the occurrence of a certain action leads to a fluent to continuously change over time, until the occurrence of a different event, finishing the continuous change. In the falling object example, the action that causes the object's height to change is that of dropping it whereas the finishing event is that of the object hitting the ground.

The common-sense law of inertia states that things tend to remain the same unless affected by external events. Applied to the falling object example, objects tend to remain at the same height unless affected by external events, as in this case would be dropping it off. The continuous change effect caused by this action makes the object's height fluent to be released from the common-sense law of inertia while it is subject to a continuous change effect. Just after hitting the ground, once again the object's height is subject to the common-sense law of inertia and it will remain at the same height point until externally affected.

Change is therefore the result of a happening action and sometimes those actions might also require a certain set of conditions to be satisfied so that the action can take place. In order to drop an object it is necessary to somehow hold the object, so therefore, if we have dropped the object is because in some way we were holding the object before the action took place. Therefore, actions and events not only need to be described in terms of the changes they happen to cause but also in terms of the preconditions that need to be satisfied so that the action can be accomplished.

Descriptions of actions and events include the descriptions of the state of the world necessary for them to take place, enumerating the so-called actions or event preconditions, and also the state of the world after they took place. Describing the posterior state of the world involves enumerating not only the instant effects but also those affected by continuous changes. As mentioned before, continuous change needs to be described along with the events that cause it and make it stop.

3.3.6 Mental States

Mental states or mental qualities (as referred by McCarthy[87]) deals with how to represent information regarding *beliefs*, *knowledge*, *free will*, *intentions*, *consciousness*, *ability*, or *wants*, which represent essential aspects of the human rationality. In his work [13] Bratman proposed the BDI (Belief, Desire, and Intentions) model for human practical reasoning, as an explanation for the human rationality exhibiting goal-driven behaviors.

Unsatisfied goals is what motivates persons to devise plans that lead to goal satisfaction, achievement or maintenance, and therefore, the emulation of intelligent behavior should inexorably be linked to an appropriate representation of the mental events involved in emulating goal-driven behaviors.

Despite the successful approaches proposed to deal with mental states, including those based on BDI models, the problem arises when it comes to dealing with beliefs about beliefs, or reason about knowledge. In exhibiting intelligent behavior, it is not only important to understand the on-going situation but also to gain knowledge about what the other agents, also involved in the situation, know about the same situation. Intelligent reactions not only depend on our knowledge about the situation but also on the knowledge held by other agents. For instance, the muddy children problem describes a situation in which a set of children are told that at least one of them have mud on their forehead. In this example, each of them can see the other children's forehead but not his/her own. To the question whether they can assume they have or have not mud on their forehead, an intelligent answer can only be supported by the knowledge that each child can see and how the rest of the children have answered the previous question. The difficulty lies here, not in describing beliefs, but in reasoning about knowledge and therefore inferring knowledge out of the knowledge about the other children's knowledge.

Some other mental qualities, such as love and hate, despite not being easy to capture and describe, they do not seem to have an important impact in emulating intelligent behavior. Similarly, with regards to emotions, Minsky in [97] defends that the way humans deal with situations and resolve problems is a direct consequence of the emotions involved in such tasks.

3.3.7 Default Reasoning

It is a fact that it is not possible to thoroughly describe all the details of a given situation. Moreover, most of this information will not be relevant and will not provide any information useful to understanding the situation. Humans are used to working under uncertainty where not all the information is known, performing what is known as default reasoning.

Adopting assumptions about unknown information is a way of overcoming lack of information. Whenever additional information appears on the scene regarding the previously made assumptions, these are revised and if any inconsistency is detected between the assumptions and the current knowledge, assumptions need to be corrected according to the new available knowledge. For example, if we are told that Tweety is a bird, it is sensible to assume that Tweety can fly. If we are later on said that Tweety is a penguin, knowing that penguins cannot fly, we can retract such an assumption and revise our understanding of the situation. As new information is being learned assumptions can be revised.

The previous sense of default reasoning falls into the so-called *atemporal default reasoning*, in which time is not involved. However, there is a different type of default reasoning in which time plays an important role. Temporal default reasoning makes two basic assumptions, that events do not have unexpected effects and that unexpected events do not take place simultaneously. The first assumption basically considers that the effects of an event or actions are those that were expected beforehand, and therefore the event or action will not cause any unexpected effect. For example, we can assume that after turning on the light switch the light will be on. However, there is a possibility of the light bulb being burned out in which case, the effect of turning on the switch will not be the light coming on. Despite the many different scenarios that might take place, it is reasonable to expect that the only effect of such action will be the light being on.

The second assumption states that whenever an event or an action is taking place it is reasonable to suppose that no additional events will take place affecting somehow expected effects of the first event. Resorting again to the light example, it can also be possible that while we are attempting to turn on the switch, simultaneously another person tries to turn it off. Since it is not possible to consider all the possible scenarios, reasoning by default provides the basis for an efficient and sensible way of performing reasoning tasks by expecting things to happens as they were supposed to unless the opposite is stated.

3.4 Requirements for Ambient Intelligence supervision

The previous section has addressed the key issues that need to be addressed by any attempt intended to automate common-sense reasoning. Additionally, and considering that most of those issues are somehow involved in the challenges faced by the Ambient Intelligence paradigm this section is intended to analyze the interrelation of both the common-sense key issues and the Ambient Intelligence requirements. It can be stated that only by providing systems with common sense can they face the Ambient Intelligence challenges. Recall that the interconnection between common sense and Ambient Intelligence has already being stated as one of our working hypotheses. This section enumerates the requirements of Ambient Intelligence and how these are strongly dependent on the common-sense key issues.

Context-awareness is one of the main requirements for Ambient Intelligence since the only way of wisely and proactively or actively reacting to context events is by understanding what is going on in the environment. The events or actions that take place in the context are noticed by means of the sensing devices deployed in the environment. Therefore, the only trace evidencing the occurrence of an event is the sensing values captured by the environmental sensors. Ascribing those values to the effects of an event or an action is the only possible way of interpretation. The accuracy in understanding context situations depends on how extensively and thoroughly preconditions and effects of events and actions have been described.

Aside from the knowledge that describes events and actions, it is also important, for the sake of coherence, to hold information about the context domain. It is a fact that sensors deployed in the environment are prone to errors and are easy to tamper with. The best way to overcome such drawbacks is by combining the information provided by environmental sensors along with the knowledge hold about the environment where such sensors are deployed. The information gathered from the environment should lead to coherent conclusions, evidencing otherwise the presence of faulty or tampered sensors. For example, suppose that the presence sensor located in a certain room is not sensing any activity. Simultaneously, the software dedicated to tracking people that analyzes the video streaming captured from the same room is tracking a moving object. It is obvious that something is not working correctly in the presence sensor since it is not noticing any presence while the people tracker is tracking, at least, a moving object. If this situation is combined with the fact that the same room has been described as a restricted access space, and therefore it is sensitive to unauthorized presence, it seems likely that a subversive action is taking place. On the contrary, if the room has been described as not subject to authorized access, it seems plausible to conclude that the presence sensor has simply stopped working.

Device malfunctioning also affects the way responses are implemented. Ambient Intelligence systems are in charge of undertaking the actions that lead the system to achieving or maintaining the context goals. Environmental devices along with the services they provide are the means that Ambient Intelligence systems can make use of to carry out the course of actions involved in the active plan. An updated list of the available devices is required so as to compose the plan on the basis of the available services. Nevertheless, device malfunctioning or unavailability is not the only challenge that need to be faced in such a highly dynamic environments, additionally, device heterogeneity need to be successfully addressed in order to make the most of all the devices present in a given context. Dealing with different protocols, architectures, or connectivity issues is the responsibility of the Ambient Intelligence system; however, the challenge here consists in understanding the set of services and the associated actions that upcoming devices can provide. For example, independently of how sophisticated a recording camera can be, it is certain that it should be capable of capturing a video sequence. If no additional information is provided by new devices, a minimum set of services and actions are associated with them.

The Ambient Intelligence paradigm works upon environments, aware of the situations that are taking place, and with the ability to generate appropriate responses to undergoing scenarios. However, on what basis does an environment conclude that a certain action is the most appropriate one? Ambient Intelligence systems, as humans, count on a set of goals to drive their behavior towards the achievement, maintenance, or performance of such environmental goals. In this respect, unsatisfied or deviated goals is what encourages the Ambient Intelligence systems to devise the most appropriate way to return to or to achieve the desired state. Rather than using hard-coded responses to whatever circumstances that might arise in the environment, it is more feasible to simply dictate the environmental goals that the system is engaged in maintaining or achieving, and try to discern among the available actions, which of them seem more suitable in reducing the distance to the unsatisfied goals.

The way to reduce the distance between the current situation and a targeted one is by devising a plan, here understood as a course of actions. Action planning is intended to consecutively apply changes to an initial state so as to transform it into the goal state. The world states notion of the action planning is very similar to that of situation proposed by McCarthy, and therefore, can be modeled by means of the possible-worlds theory. The occurrence of a given event or action produces changes in the current state of the world. Under incomplete information and reasoning by default, we can expect the world to be in a closed set of states. Action planning, therefore, consists in successively applying changes to the world state to get a glimpse of the future world state.

Plans can be therefore understood as the behavioral responses generated by the Ambient Intelligence system when unsatisfied goals arise. As mentioned before, the device dynamism and heterogeneity that characterize Ambient Intelligence makes it unfeasible to statically determine how those plans should be undertaken. On the contrary, plans should be automatically devised grounded on the knowledge of the device that is available at that moment, and the services they are able to provide.

Ambient Intelligent systems are not only expected to *react* to environmental events, but also to foresee or predict the need for a behavioral response. In a sense, the system is said to be reactive, but it also needs to be pro-active. The occurrence of some sort of situations can be preceded by the some sort of events. Whenever those situations deflect the environment from its ultimate goals, it might be desirable that the system reacts before the situation occurs, preventing the system deviation. Those undesirable situations are described as a set of preconditions, whose successive activation suggests their likely occurrence. The acknowledgment of a goal deviation should also result in the activation of some plan engaged in correcting the future deviation. For example, in a surveillance environment, a major goal of the system is that of keeping environmental integrity which at the same time involves preventing subversive actions from taking place. Whenever an unauthorized presence alarm arises in the system, independently of the behavioral response of the system, in order to deal with this situation, the likelihood of an upcoming subversive action moves the system to try to avoid it. Recall how the system is not only engaged in trying to determine the circumstances that affect the unauthorized presence, such as trying to identify the intruder, but also it is engaged in trying to prevent subversive actions from taking place. The association of unauthorized presence events to that of subversive actions relies on the fact that certain subversive actions require an unauthorized person to be present.

Independently of being pro-active or reactive responses, in order to talk about real Ambient Intelligence systems, responses have to be devised without requiring human assistance. When dealing with foreseeable situations, the only challenge is found in how to undertake the composed response, on the basis of the available services and devices. However, whenever novel situations take place in the environment that somehow requires a system reaction, analogies and goals are the only mechanisms that can support the system in devising how to respond. As mentioned above, goal deviation or unsatisfied need moves the system to undertake those actions that reduce the distance to the desired goals. Despite the fact that nothing has yet been said about analogies, it is also a powerful means that can be used in dealing when novelty. The fact that an on-going situation is considered to be unknown means that it does not match any of the situations described in the system, although it is deviating the environment from its goal or desired state, therefore requiring a response in order to redirect the environment to its previous desired state. Before exploring, among the whole set of available actions, which of those help in minimizing the distance to the desired state, it is more efficient to try to seek for analogies in the foreseen situations, and see whether adopting the same approaches works in leading the environment to the desired state.

In any case, the adopted behavioral response is determined only by the available services, in terms of time and space that characterizes the response demand. Obviously, the fact that available services at each moment in time and at each location cannot be stated in advance, limits the suitability of resorting to prefixed recipes associated to specific situations. Rather than recipes, it is more useful to describe desired states, leaving the system to instantiate the most appropriate services, among the available ones, in order to undertake the actions that lead the environment to the goal state.

Since similar situations can take place in which the same services are available, it might be convenient to learn how the system previously tackled the same situation, saving it from having to devise an appropriate solution when it already has one. These learning capabilities can also be subject to an external weighting, in order to grade solutions, letting the system choose the most valuable one when several solutions are available. Implementing learning capabilities with grading feedback not makes systems responses more instant but also more convenient, from the humans perspective.

Finally, another requirement for systems intended to Ambient Intelligence supervision is that of being capable of reasoning about knowledge. This requirement is founded on the need to gather information not only about the current circumstances that surrounds the on-going situation, but also the knowledge that the agents involved have in this regard and how that knowledge has an impact on the overall systems. The distributed nature of Ambient Intelligence systems might also lead to knowledge distribution yielding the need for reasoning on the basis, not only of the information gathered, but also on the basis of what other entities know about that situation. The agent theory is closely related to that requirement, since as will be justified later on, a Multi-Agent System is the most suitable approach to gather context knowledge and to reach the environmental goals. This requirement therefore refers to the agent's capacity for reasoning about other agents' knowledge.

3.5 Benchmark problems for understanding purposes

The lack of benchmark problems for common-sense reasoning in Ambient Intelligence contexts makes it unfeasible to compare the proposed approach with existing systems. For evaluation purposes, it is essential to postulate appropriate benchmark problems disclosing the benefits of a common-sense-based approach in comparison to the systems proposed to date.

Due to the many aspects involved in common-sense reasoning, it is not suitable to devise a single benchmark problem considering all the aforementioned key issues. On the contrary, it is more appropriate to design very specific benchmark scenarios in which only one key issue comes into play. The idea behind this proposal is to pose situations that, only by holding common sense capabilities, can be understood. These benchmark problems are not yet dealing with behavioral response generation, but rather, they are targeted on demonstrating the importance of common sense in context understanding. A benchmark problem is proposed for each of the key issues stated in the previous section.

The majority of the systems intended to Ambient Intelligence supervision have traditionally been deployed to home assistance. In fact, Home Assisted Living (HAL) is one of the hot topics of intelligent systems. For that reason, the benchmark scenarios presented here are taking place in a kitchen of a HAL, so that the performance of the existing solutions can be compared.

- **Effects of events:** Light detection after the light switch has been switched on has to be interpreted as though the effect of turning on the light is light detection.

- **Context-sensitive effects of event:** Since kitchens are designated places for cooking, momentary temperature growths are direct effects of cooking events.

- **Non-deterministic effects of events:** Suppose there is water on the floor and someone is stepping on it. If the person does not pay special attention when walking, the effect of stepping in water might be the person slipping.
- **Delayed effects of events:** If the tap is turned on and the stopper is put in the sink, after a while, the water will have reached the sink height and the water will start to spill onto the floor.
- **Canceling effects:** The event of opening a door by means of a door actuator should result in the door being opened. However, if the door does not open when the actuator is actioned, then it can be inferred that a canceling event might have taken place. For example, blocking a door with an object heavier than the momentum that the actuator can apply, causes canceling effects upon the opening event.
- **Cumulative effects:** When more than one kitchen hob is on, the temperature increases more quickly. Therefore, if the temperature growth is known when a kitchen hob is on, higher temperature when additional hobs are on can be associated to the cumulative effects of several kitchen hobs turned on.
- **Concurrent events:** Since people cannot enter two different rooms of the same house at the same time, the activation of the presence sensors of two different rooms of a single resident house needs to be interpreted as more than one person being present in the house. It might also have additional implications depending on the time of day, since it might be due to a relative's visit or it might be caused by a home break-in.
- **Indirect effects of events:** If the kitchen light suddenly burns out while one is in the kitchen, it is desirable not to crash head-on into any of the kitchen furniture. In order to safely leave out the kitchen an alternative lighting source should be found.
- **Reasoning about space:** if the kitchen light sensor detects light, and there is no kitchen window and the door is closed it must be due to an external source of light being on. In the kitchen, two sources of light are the cooker hobs and the inside of the fridge.
- **Common-sense law of inertia:** If doors and windows of a room are closed these are expected to remain closed unless externally affected. In the same sense, if the water level is increasing, in the scenario of the sink with the stopper in, it is expected to increase unless affected by an external event, like turning off the tap.
- **Continuous change:** When a person is walking, tracked by a system, the person's position is a property subject to continuous change while the person is walking. The trajectory of a person, determined by the position sequence that has been followed by the person walking, can also be used for predicting future positions, on the base of continuous change.
- **Mental states:** We can prevent a person from slipping by warning him about the wet floor. A wet floor is therefore understood as a precondition for a person to slip. Since one of the system's goals is to preserve the resident's physical integrity, it is sensible to understand a wet floor as a risk that might deflect the system from its target goal, which is to preserve the resident's physical integrity.
- **Default Reasoning:** Resorting again to the open faucet scenario, it can now be assumed that it is opened because something needs to be soaked and that the stopper is not in. However,

> when the water level starts increasing and it finally results in the water being spilled onto the floor, the assumption can then be revised so as to consider that it might have been left on by mistake and that the sink stopper was also in. Regarding atemporal default reasoning, we can also resort to the kitchen light scenario. After switching on the light, this can be expected to be turned on. The lack of light means that something did not go as expected and the light did not turn on, probably due to the bulb being gone.

This benchmark scenario depicts a set of situations which, unless previously considered, systems lacking common sense will not be able to understand. Further details about how different state-of-the-art systems deal with these scenarios will be provided in the Validation chapter.

3.6 Interim conclusions

This chapter has been mainly dedicated to presenting the essential aspects of common-sense reasoning and how those have an impact in the context understanding task demanded by any system intended for Ambient Intelligence. The enumeration of the issues has been grounded in the work of Mueller, Hayes, and Minsky. Individually, they have focused on key issues of common sense although from different points of view; Mueller pays attention to issues of essential form reasoning tasks; Hayes is especially concerned with the physical aspects of the worlds, while Minsky's work intends to provide an explanation about how the human mind works. Combining these three perspectives, we have been able to reduce these different views to the quintessence of common-sense, listed in the key issues for common sense section. Moreover, the need for these requirements has been justified by resorting to those stated by the Ambient Intelligence paradigm.

This chapter has also surveyed the state-of-the-art of systems for automating common-sense reasoning. Due to the fact that those systems are explored from the perspective of the Ambient Intelligence paradigm, those features specifically designated to support an autonomous and intelligence behavior are more carefully considered. On the basis of that comparison, it can be concluded that Scone is the system that better fits the requirements of an approach intended to be deployed in an Ambient Intelligence environment.

Finally, due to the fact that benchmark problems for common-sense reasoning have not been specially designated for Ambient Intelligence, this chapter also takes the responsibility of proposing a set of scenarios where the common-sense key issues come into play. These benchmark problems will be supplied to those state-of-the-art systems for Ambient Intelligence that have obviated the need for common sense. These results will be presented in the validation chapter.

Chapter 4

Modeling and Reasoning About Context

Whenever we write an axiom, a critic can say that the axiom is true only in a certain context. With a little ingenuity the critic can usually devise a more general context in which the precise form of the axiom doesn't hold.

– *John McCarthy in Generality in Artificial Intelligence*

Summary – *It is obvious that context plays an essential role in Ambient Intelligence, not only in determining the set of activities that take place in it, but also in devising the most appropriate response to those situations. Context is also essential for disambiguating knowledge and meaning. This chapter is therefore devoted to analyzing the most compelling approaches for modeling and reasoning about context.*

4.1 Introduction

The notion of context is at the heart of the Ambient Intelligence paradigm because of its role in narrowing down the meaning of the environmental events and in determining suitable means to react to undesired situations. Despite the importance of context, this concept has not yet been universally formalized. On the contrary, the fact that the notion of context is a relevant issue for different fields of knowledge such as natural language understanding, linguistics, context-awareness, or knowledge representation among others, makes it difficult to provide a common and unique definition of what context is. Some authors such as McCarthy [91] echo this peculiarity, deciding not to offer a definition of context since, from their perspective, it is as pointless as asking about the definition of a group element [3].

Rather than trying to provide a definition of context, John Sowa in [136], distinguishes three different functions for the notion of context: collecting the syntax of a given context; semantically mapping linguistics to physical situations; and pragmatically concerning the convenience or the purpose for a context to be considered in isolation. The confusion therefore lies in using the notion of context as though it simply refers to one of these functions.

In addition to its three-dimensional character, the notion of context when considered under the perspective of Ambient Intelligence pursues a twofold aim. On the one hand, the context notion should encompass the required information for recognizing and understanding undergoing situations. On the other hand, the context notion should also be devised as a set of devices and services accounting for the environmental acting capabilities. Context is then expected to implement responses to environmental situations by means of the tools available to it.

Traditionally, these responses have been devised as static recipe reactions that trigger whenever context information seems to match any of the considered situation patterns. In this regard, the majority of the contributions found in the literature make the assumption that the systems will only be facing previously considered scenarios, thereby overlooking unforeseen ones. On the contrary, even the scenarios described in [35], where the concept of Ambient Intelligence was firstly proposed, were also being constrained to dealing with those scenarios that fell into what one might reckons as *normality*. But, how are those systems then expected to behave when context does not evolve as expected? Or in other words, how are they expected to handle *novelty*?

Before being able to provide a solution to these questions, it is necessary to devise a mean to capture and model the semantics of "*normality*", extending it to consider "*abnormalities*". Partially, this endeavor has already been addressed by the *context-awareness* paradigm [128]. Context-awareness is referred as the system capability that allows it to gain knowledge about its surrounding and adapt their behavior to autonomously act on behalf of users. Nevertheless, the lack of consensus reflected in the multiple meaning of context has also affected the context-awareness theory. Motivated by the need to provide a domain-independent notion of context, Dey and Abowd in [1] reaches a wide consensus on their definition and categorization of context and context-aware computing. However, according to the Sowa's work [136], rather than consolidating an integral definition of context, the work of Dey and Abowd basically contributes to consolidating the syntactic dimension of the context. From the perspective of context-awareness and Ambient Intelligence, the semantic and pragmatic dimension of the context notion seems to be relegated to a second position, since, to the best of our knowledge, they have not been directly addressed by any of the approaches found in the literature.

Aware of these shortcomings, this thesis is intended to provide a comprehensive approach for context modeling that encompasses the three dimensional view of context and the twofold aim that characterizes it when considered from the perspective of Ambient Intelligence. Furthermore, this holistic approach should also take into account that context modeling is not only restricted to considering expected situations, but rather it should be enhanced with the required means for recognizing and understanding unpredicted or unforeseen scenarios.

The remainder of this chapter is structured as follows. First, a revision of the state of the art for context modeling and reasoning is presented. The next three sections are concerned with the description of the three dimensional views of the context concept: syntax, semantics, and pragmatics. Section 5 describes the proposed approach for characterizing the different types of situations that might take place in a context: normal, abnormal, and unknown. Finally, Section 6 presents the conclusions drawn from the proposed approach.

4.2 Previous work

A context model for Ambient Intelligence comprises the rules that establish how to map sensor data values onto high level knowledge. These rules, far from being unique and common to context-aware systems, tend to be tailor-made solutions that prevent context-aware systems from sharing and leveraging the knowledge they hold. Context models are therefore characterized for their lack of interoperability.

The work in [11] provides an appropriate starting point for surveying the existing modeling techniques. In [11] authors split context into three levels of granularity. The first level deals with raw sensor data, the second level is concerned with interpreting those data, as they all provide knowledge about an on-going situation, and finally, the third level surveys works that extract information by establishing relationships among the situations of the second level. Despite the fact that much attention is paid to what Sowa termed the *syntactic dimension of the context*, little is paid to the *semantic* and none to the *pragmatic* dimension.

The first of the surveyed approaches is Context Modeling Language (CML) [60] [58] [59], which on the basis of a database modeling technique is intended to capture the concepts that are present in the context, along with the relationships established among them. CML claims to provide support for reasoning, however, rather than doing so, one might reckon that it simply consists in answering SQL-like queries. Inferences or deductions are therefore outside the scope of the functionality provided.

Context information can also be modeled with regard to the location, in a geo-spatial sense. The approach in [47] proposes different conceptualizations of the world by constructing a multilevel model of the world. The model proposed within the Nexus project consists in the Augmented World Model [102]. This proposal adopts an object-based approach in which multi-inheritance is supported by the context objects. It has to be highlighted that this model was motivated by the need to overcome the interoperability issues that characterize context-aware applications.

Similarly, the Equator project [53] models the context by means of an OWL class model in which ontology entities are symbolic spaces, arranged in a hierarchical fashion and whose properties represent the relationships established between these spaces. In this sense, as might be inferred from revising the context modeling state of the art, the most commonly used approach is that of ontologies, especially, those based on OWL.

The main strength of an ontological approach is a direct consequence of using a standard language, such as OWL. This language provides support for interperoability and information sharing. Additionally, a more restricted version of OWL language, as it is OWL-DL performs well when it comes to reasoning about the context knowledge. Some examples of successful frameworks for pervasive applications that have resorted to this approach are CoBrA[20], SOCAM[54], SOUPA[21], or Gaia[115].

As one might notice, the works presented so far have only been concerned with knowledge that could be directly inferred from the raw sensor data. However, the semantic level goes one step further and engages in interpreting or understanding the scenario that is taking place in the considered context. Although aimed at the field of meaning in natural language, the theory of *situation semantics*, proposed by Barwise and Perry [10] has been extrapolated to context-awareness. However, as stated in [136], situations cannot be completely described by propositionally enumerating all the aspects involved in the situation since aspects such as intuitions about context escape this modeling strategy.

Sowa also proposes his own theory [135] for context modeling, based on conceptual graphs of semantic networks. Under this theory, contexts are modeled as propositional containers of additional conceptual graphs. In [134] Sowa proposes the use of context so as to split knowledge bases into small pieces of knowledge. Note that Guha in [56] founds his work on that premise, and from there he proposes the notion of microtheory as a means of organizing knowledge in Cyc [77].

McCarthy's *ist(c, p)* predicate [90], which can be read as "*proposition p is true in context c*" was his attempt to provide a universal mechanism to overcome the large number of logics arising for different reasoning theories. McCarthy's theory was also implemented in Cyc by his student Guha [56]. However, in spite of its great success with regard to supporting the construction of a large knowledge base, the distinction between syntax and semantics is not clearly stated in the context model. As stated by Sowa in [136] the *ist* proposition mixes these two dimensions, in the sense that the *ist* predicate holds the meaning of the proposition p being contained in context c but also, the semantic dimension of proposition p to be true in context c. A possible way of decoupling both dimensions is to resort to the Barwise and Perry's notion of *situation*, already mentioned above. As stated in [137], it is possible to establish a connection between a given proposition and a certain situation, in such a way that all the propositions encompassed in that situation are considered to be true.

Additionally, the meaning of these propositions is unavoidably associated to the context in which they are being considered. In this sense, *meaning* is expected to be something more elaborate than just mere conventions about what other concepts state their significance to be. On the contrary, the meaning associated with a proposition has to be given in terms of how it affects the context. A plausible way of doing so is by means of the "*possible world*" theory. As stated in [112] there are two possible ways of describing what a *possible world* is. On the one hand it can be described as a set of consistent propositions that are true in a given *world*. On the other hand, a *possible world* can also be explained as an account for how things can be interpreted in a given *world*. The same author states the following, regarding the existing relation between context and possible worlds:

> " *contexts are possible worlds in which judgments are derived, so that each judgment stated true by the theory is true in a certain world, namely the one providing all the informational data needed in order to acquire the knowledge contained in that judgment (i.e. the information which expresses the conditions to verify the propositional content of the given judgment). This world is namely expressed by a context*[1]."

Later on this chapter, in Section 4.5 there is a description of how possible worlds can be formalized by means of the Kripke structures.

4.3 The context syntax

After having reviewed the state of the art for context modeling, this and the following sections are devoted to describing the insights of the proposed strategy for modeling Ambient Intelligence contexts. To this end and, inspired in the three-dimensional view of the context notion advocated by Sowa, the description of the proposed modeling approach is organized in terms of the dimensions previously described.

Consequently, this section concerns capturing the context syntax, and to this end, the lexicon of the proposed context model has to be provided, along with the rules that determine how these symbols can be combined, by means of a context-free grammar.

The identification of the context lexicon starts by establishing all the categories or entities comprising the vocabulary of the proposed language. The revision of some of the most successful and widely accepted categories for context-aware systems, such as the ones proposed by Ryan in [122], Schilit in [129], or Dey and Abowd in [1], leads to the conclusion that despite being right in the insights, the proposed categories fail to comply with the requirement of low-coupling between the different context dimensions. Whereas categories such as *location* and *time* are part of the lexicon, some others such as *activity* or *identity* escape from the syntax dimension and should be part of the semantic one. However, this poses a question; the identification of what should be part of the lexicon and what should be kept outside it? For the sake of preserving the independence of the three context dimensions the proposed approach adopts the convention of considering that just the information directly extracted or related to sensors should be considered part of the lexicon.

The fact that Ambient Intelligence systems are fed with raw information, directly gathered from sensors, simplifies the vocabulary lexicon needed to describe such contexts. In contrast to what might happen in Natural Language Processing, in which the considered vocabulary must be the whole language itself, in Ambient Intelligence, vocabulary is limited to those items that handle actions and events and those that contain the sensed environmental data.

---

[1] PRIMIERO, G., Information and Knowledge. A constructive Type-theoretical Approach.Logic, Epistemology, and the Unity of Science, Vol. 10. ISBN 78-1-4020-6169-1, pp. 152

The context syntax is intended to strictly enumerate the domain concepts that are considered in the modeled context. High level knowledge is therefore left for the upper dimensions (semantics and pragmatics), preserving the low-coupling requirement between this and the remaining context dimensions. The main advantage of achieving a non-coupling model is to do with the benefits of addressing each dimension with the modeling approach that best fits the features of each dimension.

Regarding the syntactic dimension, the modeled approach basically consists in describing the symbols and the rules that determine how these symbols can be combined. Despite the fact that OWL or OWL-DL have traditionally been chosen for modeling the context syntax, more basic and efficient mechanisms can also be used to cope with the task of capturing and modeling the syntactic dimension. Recall Sowa's definition of the syntactic dimension which only ascribes to it the responsibility for enumerating the symbols that are required for expressing knowledge about contexts, along with the rules that determine how those symbols can be combined. At this level nothing has to be said about the meaning of those symbols or their relationships. These are the responsibility of the upper context dimensions, and therefore, they will be addressed in the following sections.

Similarly to how lexicon of formal languages is established and bearing in mind both the need for simplicity and efficiency restrictions, the proposed approach for modeling the syntactic dimension of context follows the theory of formal language.

Using the notation and definitions of this theory, a context-free grammar is postulated as a means of capturing the context syntax. The following definitions present the foundations of a formal language for modeling Ambient Intelligence contexts.

Definition 1: A context-free grammar is defined as a four-tuple, such that:

$$G = (\mathcal{B}, \mathcal{E}, \Gamma, S_0) \tag{4.1}$$

where:
S_0: is the initial axiom or symbol.
$\mathcal{B}$: is the alphabet or lexicon. It is also known as the set of terminal vocabulary of the grammar.
$\mathcal{E}$: is the set of non-terminal symbols of the grammar.
Γ: is the set of production rules, such that:

$$\Gamma : \mathcal{E} \rightarrow X_1, \ldots, X_n \text{ where } X_i \in (\mathcal{B} \cup \mathcal{E})^* \tag{4.2}$$

Definition 2: $\mathcal{L}$ is the language such that $\mathcal{L} = L(G)$, whose grammar $G = (\mathcal{B}, \mathcal{E}, \Gamma, S_0)$ is defined as follows:

$\mathcal{E} = \{S_0$, Definition, Predicate, Statement, Type, Device-id, Service-ID,
Action-ID, Object-ID, Event-ID, Place-ID, Time-ID$\}$

$\mathcal{B} = \{$event, action, device, service, object, place, value, time, provides,
performs, at, in, upon, has-value, is-a, causes, identifier $\}$

$\Gamma = \{$
(1) $S_0 \rightarrow$ Predicate in-context C
(2) Predicate → Definition | Statement
(3) *Definition* → identifier is-a Type
(4) *Statement* → identifier has-value value
(5) *Statement* → Device-ID provides Service-ID

(6) *Statement* → Service-ID performs Action-ID
(7) *Statement* → Action-ID upon Object-ID
(8) *Statement* → Event-ID at Place-ID
(9) *Statement* → Event-ID in Time-ID
(10) *Statement* → Device-ID causes Event-ID
(11) *Device-ID* → device
(12) *Service-ID* → service
(13) *Action-ID* → action
(14) *Object-ID* → object
(15) *Event-ID* → event
(16) *Place-ID* → place
(17) *Time-ID* → time
(18) *Type* → event | action | device | service | object | place | time
}

At this level, Ambient Intelligence contexts are characterized only in terms of statements that describe the devices and services deployed in them. The following examples show valid statements of the language that correspond to a simplified description of a room in which a presence sensor is deployed:

```
sensor is-a device
presence-sensor-1 is-a sensor
room-1 is-a place
presence-sensor at room-1
presence-sensor-service-1 is-a service
presence-sensor-1 provides presence-sensor-service-1
detection is-a action
presence-sensor-service-1 performs detection
moving-object is-a object
detection upon moving-object
current-time-instant is-a time
event-detected-presence-1 is-a event
event-detected-presence-1 at current-time-instant
presence-sensor-1 causes event-detected-presence-1
```

Note how nothing has yet been said about the meaning of any of the statements or the concepts used above. At this level the sole responsibility of a hypothetical *context manager* utility is to gather the raw sensor data and to express that information by means of the proposed language statements. The context manager functionality has been implemented by means of an intelligent agent, and therefore, it will be described in Part III, as part of the acting capabilities.

4.4 The context semantics

The semantic dimension of the context notion, built upon the syntactic one, concerns the identification of the situations that are referred by the statements of the lower layer. Inspired by the work of Knuth in [72], this thesis has sought to assign meaning to the lexicon vocabulary by connecting these symbols to entities of a common-sense knowledge base, instead of associating attributes to such symbols.

Common-sense knowledge can be described as the common knowledge, held by humans, that explains how the "*the world works*". In this regard, the impact that capturing such knowledge has on achieving intelligent systems was long time ago envisaged by McCarthy and Minsky, as related by the latter in [95]: "*In 1959, John McCarthy came to MIT from Dartmouth, and we started the MIT Artificial Intelligence Project. We agreed that the most critical problem was of how minds do common-sense reasoning. McCarthy was more concerned with establishing logical and mathematical foundations for reasoning, while I was more involved with theories of how we actually reason using pattern recognition and analogy*"[2]. Since then, there have been several attempts to build a knowledge-base to manage the vast amount of information involved in formalizing common sense. As mentioned in Chapter 3, among the most promising approaches to building common-sense knowledge bases it is worth mentioning the efforts of Cyc [77], WordNet [43], or Scone[2].

Building the semantic dimension of a context modeling approach is a task that has to be supported on top of a knowledge-based system. The devised knowledge-base should support not just semantic ascription to syntactic knowledge but also higher level functionalities such as reasoning, deductions, or inferences. On the basis of these requirements, this thesis resorts to a common-sense knowledge-base system that not only captures the context semantics but also leverages human-like reasoning capabilities.

Based on the syntactic dimension, in which concepts and rules have been enumerated, the semantic dimension is captured and formalized by means of a semantic model for actions and events. Considering that Ambient Intelligence contexts are mainly intended to supervise ongoing events and generate behavioral responses to those events, the notion of *action* and *event* should occupy a central role in the proposed model. Before addressing the computational aspects of how the semantic model should be formalized, an analysis of how the philosophical discussion has addressed the representation of actions and events is necessary.

Actions and events have commonly been treated as equivalent, or as having the slight difference of considering actions as events which have been intentionally generated [62]. On the contrary, the theory of action for multi-agent planning [50] advocates a distinction between actions and events, although it hints that actions are accomplished by agents in their endeavor to achieve a goal.

Davidson's theories, particularly those regarding the philosophy of action, also identify actions with events, as argued in [30]. Actions are described as a combination of two views. On the one hand, actions can be seen as causal explanations of body movements and on the other hand, they can also be seen as the justifying reason that leads the action to take place. Davidson considers events to be equivalent to actions. The sole difference is that when an action is considered as an event, it is re-described in terms of its effects.

The model proposed here for actions and events adopts the Davidsonian view. It should be highlighted that Cyc [77], through its language CycL, represents actions and events using a Davidsonian approach. Actions are described as events but are carried out by an agent. The approach implemented in Scone has been extended to include the notion of primary reasons for an action, along with its temporal and location aspects.

Apart from the concept of action and event, some other relevant entities must also be considered in relation to actions and events so as to capture their semantics. The following definitions state the foundation of the proposed model for actions and events:

Definition 1. A Context is a set $\mathcal{C}$ composed of statements which, when used together, describe the knowledge about the world. There may be multiple contexts describing each of the different views of the world. The meaning or truth value of a statement is a function of the context in which it is being considered.

The function *meaning* : $\mathcal{T}, \mathcal{C} \rightarrow \mathcal{M}$, where $\mathcal{T}$ is the set of statements describing the world, $\mathcal{C}$ is

---

[2] http://www.cs.cmu.edu/ sef/scone/

the set of possible contexts, and $\mathcal{M}$ the set of possible meanings, $meaning(s,c)$ therefore returns the meaning or truth value of the statement s in the context c. This can be formally stated as:

$$\forall c_i \in \mathcal{C} \forall s_i \in \mathcal{S} : m_i = meaning(s_i, c_i) \iff s_i \subseteq c_i \tag{4.3}$$

The meaning or truth value of a given statement depends on the contexts in which it has been declared.

Definition 2. An Action $\mathcal{A}$ is causally explained from the perspective of its relation to the primary reason that rationalizes it. The function $AG : \mathcal{A} \rightarrow \mathcal{G}$, such that $\mathcal{A}$ is the set of possible actions and $\mathcal{G}$ is the set of possible agents, returns the agent performing the given action. Furthermore, the function $PR : \mathcal{A}, \mathcal{G} \rightarrow \mathcal{E}$ is the primary reason for an agent performing an action to seek the effects of the event caused. Finally, the function $PA : \mathcal{A} \rightarrow O$, such that O is the set of possible objects, returns the agent that performs the action upon the given object.

$$\exists g \in \mathcal{G} \exists a \in \mathcal{A} \exists o \in O : (AG(a) \wedge PR(g,a)) \iff PA(a,o) \tag{4.4}$$

Therefore, an action is performed upon an object, if and only if, there exists an agent with a primary reason to perform the action.

Definition 3. An Event $\mathcal{E}$ is the individual occurrence that causes changes in the world. The criteria followed by the Davidsonian doctrine on individuation of events argues for the equality of events when the same effects occur. The Davidsonian view is here adapted to internalize the multiple contexts approach. In this paper it is therefore considered that two events are equivalent when the same effects are caused by different actions. The effects of events are captured in the *after context*, while the preconditions for an event to take place are described by the *before context*. The functions $BC : \mathcal{E} \rightarrow \mathcal{C}$ and $AC : \mathcal{E} \rightarrow \mathcal{C}$, such that $BC(e)$ and $AC(e)$, respectively return the statements of which make up the before and after context of a given event. Furthermore, the function $effect : \mathcal{A}, O \rightarrow \mathcal{S}$, returns the set of statements that describe the world after the action $\mathcal{A}$ took place upon object O.

$$\forall e \in \mathcal{E} : (BC(e) \cup effect(a,o)) \rightarrow AC(e) \tag{4.5}$$

Given the events e_1 and e_2, it can be said that e_1 is equivalent to e_2 when they have equivalent *after contexts* or when they cause the same effects:

$$\exists e_1, e_2 \in \mathcal{E} : e_1 = e_2 \iff AC(e_1) \subseteq AC(e_2) \tag{4.6}$$

Definition 4. A Service $\mathcal{S}$ is provided by a device $\mathcal{D}$ and it performs a set of actions upon an object or a set of objects. The function $PD : \mathcal{S} \rightarrow \mathcal{D}$, such that $\mathcal{D}$ is the set of available devices, returns the device or devices that provide a given service.

$$\exists s \in \mathcal{S} \exists d \in \mathcal{D} \exists a \in \mathcal{A} \exists o \in O : (PA(a,o) \wedge PD(s)) \rightarrow AG(a) = d \tag{4.7}$$

The definition of service therefore implies that the agent of an action provided by a service is a device.

Definition 5. An Object is the set O of possible environmental objects upon which actions are performed. The function $OA : \mathcal{A} \rightarrow O$ returns the set of possible objects that can receive a given action.

$$\exists o \in O \exists a \in \mathcal{A} \exists e \in \mathcal{E} : OA(a) \wedge PA(a,o) \rightarrow e \tag{4.8}$$

The occurrence of an event e implies the existence of an object o upon which the action a is performed.

4.5 The context pragmatics

Both semantics and pragmatics concern meaning, however, semantics assumes that there exists a precise meaning for every concept, while pragmatics goes one step further and considers how that meaning may vary depending on the surrounded circumstances [63].

Humans can hold multiple meanings of a concept, even inconsistent ones, with little effort. For example, humans do not find any inconvenience in concurrently holding the propositional knowledge that states the fact that Bill is a dog and that other stating that there also exists a person named Bill. Saying now that *Bill barks* is obviously a statement that is referring to Bill, the dog. Not so obvious is the following fact: *I told Bill to stop barking at me, it was not my fault*, but even then, humans would easily identify that the Bill referred here is the person.

The logical or computational representation of both facts finds an incongruence since Bill cannot be a dog and a person at the same time. On the contrary, people do not seem to have problem in dealing with this sort of information. Therefore, this situation poses the following question: how do people manage to deal with such an incongruent knowledge? Philosophers have pointed to the theory of "*possible worlds*".

The meaning of the theoretical concept of "*world*" is used to analyze a set of key concepts, so-called "*worldmate concepts*" [34], from their respective domains [14] and preserving the existing meaning differences. One world could be used to represent Bill, with Bill being a dog, while some other world could be used to represent Bill, now as a person. Either worlds are plausible although incongruous, however, this way of representing information as isolated worlds, enables the representation of inconsistent information in a logically consistent manner. This is how humans are capable of reasoning and making inferences about different situations, some of which might be inconsistent with previous knowledge. Humans only consider those worlds that are more plausible or typical on the basis of the knowledge they hold at that moment [142]. This is also the way that permits humans to easily deal with the vast amount of knowledge that makes up common sense.

Addressing modality has traditionally been the primary target of the theory of possible worlds [34]. The truth value of a certain proposition is evaluated in terms of the proposition being possible, impossible, necessary or contingent. However, the possible worlds theory can also be used for modeling the pragmatic dimension, as stated in [111].

Possible worlds can be interpreted as both, the set of sentences evaluated to true in a given context, or in relation to the signification assigned to a given context. In either case, a possible world comprises the constitutive sentences, in an ordered manner, as though they were part of a knowledge process. This idea is supported in the Kripke structures [74], devised as a means to formalize semantics in terms of possible worlds.

Definition 6. Context semantics can be described by means of the Kripke models such that:

$$M = (S, R, \Pi)$$

in which S is a nonempty set of states or possible worlds, R is the relation between the possible worlds, and Π is the function that tells us which propositions or statements are true in each of the possible worlds, such that:

$$\forall s_i \in S, \Pi(s) : \Theta \rightarrow \{\text{true}\}$$

where Θ is the set of propositions or sentences that are true in the possible world s_i.

Time and location are the two main sources of incongruency when asserting knowledge in a knowledge base. For example, the fact that represents the cabinet door being opened is incongruent with another fact that states that the door is closed. Obviously, someone can tell that attaching the time-stamp to those occurrences avoids the incongruence problem without requiring possible worlds. However, what if instead of a cabinet door, the fact to be modeled is that one in which the stopper is put in the sink and the faucet is turned on. Modeling the state of the world after these two actions take place requires a more expressive mean than just two time-stamped facts stating that the stopper is put in the sink and that the faucet is turned on. If a person is asked to model the state of the world after these two actions take place, s/he will depict a world in which the water level in the sink will be increasing until the sink height is reached, then, water will start spilling onto the floor. This is the sort of situation that cannot be modeled by simple statements in a knowledge base, therefore demanding more expressive mechanisms.

Common sense is what enables people to perform non-trivial inferences, such as those involving delayed effects of actions, like the spilling water scenario. Similarly, the proposal presented in this thesis is grounded in incorporating such common-sense knowledge into the pragmatic dimension, using to this end an approach based on the possible worlds theory. This approach can be formalized by means of the aforementioned Kripke structures. This can be best illustrated by a simple example, making the most of that Kripke structure property that allows its representation using the graphs, where nodes represent states or possible worlds, with their corresponding propositions, and arrows depict the connections between those worlds.

The pragmatic dimension is build upon the semantic dimension and therefore, the Kripke model proposed here is grounded in the semantic model presented in the previous section. Action and event concepts of the semantic model are here imbued with the pragmatism of the possible world semantics. Both actions and events are described as though they were knowledge processes where milestones are used to separate the enclosed worlds. Those actions and events, labeled here as basic actions[3], will do with just two worlds, one for the context before the action or event takes place and another for the context afterwards.

Let the Kripke model for the *turn on faucet* action be:

$M_{turn.on.faucet} = (S, R, \Pi)$
$S = \{s_0, s_1, s_2, s_3, s_4, s_5, s_6\}$
s_0 = before turning on the faucet with empty sink and stopper put on
s_1 = before turning on the faucet with empty sink
s_2 = before turning on the faucet with non-empty sink
s_3 = after turning on the faucet without having put the stopper on
s_4 = after turning on the faucet level-of-faucet-drain equals (flow*(elapsed-time/ base-area)
s_5 = after turning on the faucet with water level having reached the sink height
s_6 = after turning on the faucet with faucet-liquid being dropped-off

---

[3] Those which do not involve fluents. Fluents are time-varying properties of the world, where more than the traditional milestones (*before* and *after worlds*) will be identified.

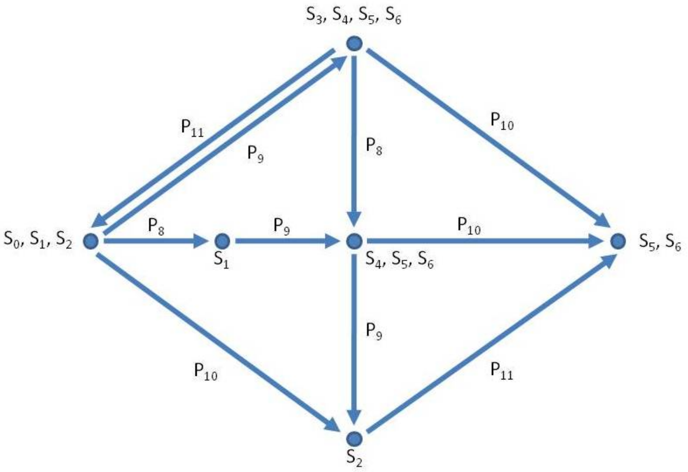


Figure 4.1: Kripke model for the *turn on faucet* action

$\Theta = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}\}$
$p_0 \rightarrow$ faucet valve is turned off
$p_1 \rightarrow$ faucet valve is turned on
$p_2 \rightarrow$ stopper is put on the drain
$p_3 \rightarrow$ stopper is not put on the drain
$p_4 \rightarrow$ the sink is empty
$p_5 \rightarrow$ level of faucet liquid contained in the sink is non-empty
$p_6 \rightarrow$ level of faucet liquid contained in the sink is full
$p_7 \rightarrow$ liquid contained in the sink is being dropped-off
$p_8 \rightarrow$ put the stopper on the drain
$p_9 \rightarrow$ turn on the faucet
$p_{10} \rightarrow$ level of faucet reaches the sink height
$p_{11} \rightarrow$ turn on the faucet
$\Pi(s_0) = \{p_0, p_2, p_4\}$
$\Pi(s_1) = \{p_0, p_4\}$
$\Pi(s_2) = \{p_0, p_5\}$
$\Pi(s_3) = \{p_1, p_3, p_4\}$
$\Pi(s_4) = \{p_1, p_2, p_5\}$
$\Pi(s_5) = \{p_1, p_2, p_5, p_6\}$
$\Pi(s_6) = \{p_1, p_2, p_5, p_6, p_7\}$

From the description of the *turn on faucet* action we can easily construct a Kripke structure that describes how possible worlds are interconnected on the basis of occurring events. Figure 4.1 depicts the Kripke structure for such scenario. Each structure node is labeled with the set of worlds that are considered possible given the previous state of the world and the events or actions occurring. The structure arrows represent the occurrence of events that trigger migrations to different possible worlds.

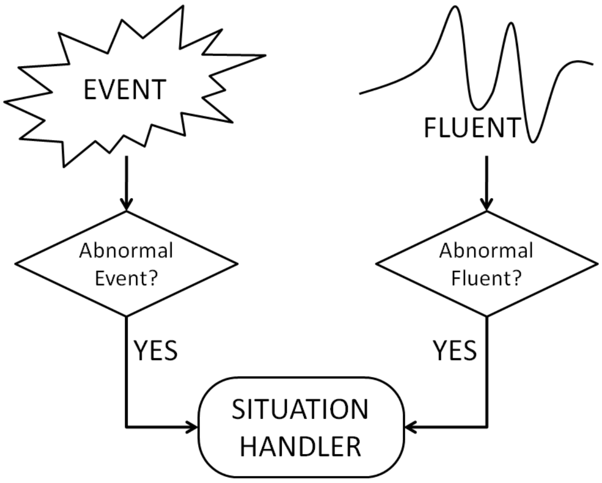


Figure 4.2: Overview of the process for abnormal situation identification

4.6 Situation characterization

One of the main challenges of context-aware systems is that of having to understand and recognize context activities with the sole information of sensor data. This complexity manifests itself when having to model raw sensor data in such a way that not only isolated actions can be spotted but also higher level combinations of them, referred to here as *activities* or *situations*. To this end, approaches originally conceived for fields such as human activity or pattern recognition can be adapted to work in situation understanding and recognition systems. Thinking of activities or situations as sequences of actions that occur at a given location and at a specific time, entails a primary identification of the involved individual actions, ignoring time and location. Secondly, those actions are considered globally so as to recognize a pattern of an ongoing situation or activity.

Among the different activities or situations that may take place in the context of Ambient Intelligence special attention is paid to identifying *abnormal* situation. Objectively, situations cannot be labeled as *abnormal* in isolation. On the contrary, situations need to be contextualized in order to determine if they match any of the abnormal situations to which the Ambient Intelligence context should be susceptible. Commonly, these situations, which do not fall into what one might reckon as a normal situation for the considered context, demand some sort of response from the system in charge of supervising the environment. It is therefore particularly important to devise a mechanism that supports the system in the task of identifying those situations that can be potentially abnormal.

The proposed approach for identifying a potential abnormal situation is grounded in a double strategy, as depicted in Figure 4.2:

1. Some particular events suggest the occurrence of abnormal situations.
2. Properties that change over time (*fluents* in the Event Calculus) can be used to characterize abnormality.

There are some situations that can be characterized by the occurrence of certain events, in such a way that the single occurrence of one of those events provides likely evidence that the situation

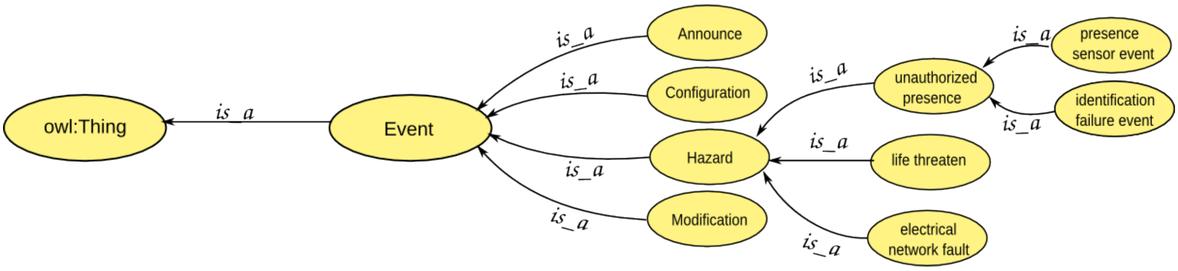


Figure 4.3: Extract of the taxonomy for an abnormal event characterization.

taking is place. Therefore, the first strategy consists in characterizing those events, in a sufficiently abstract manner, so that their occurrence can be identified under any circumstances. For example, an unauthorized presence alarm might suggest that a subversive action is taking place in the context. The *unauthorized presence* event has to be characterized in such a way that it matches different types of events, such as for example the triggering event of a presence sensor located in a restricted access room, or the event that performs some sort of biometric identification resulting in a failed identification.

However, it cannot be established that there exists a direct relation in between the occurrence of these events and the existence of an unauthorized presence because it might be possible that they are due to human errors, different context conditions, or system errors. It is therefore necessary to discern between any of them, and for that reason, whenever any of these events is detected further analysis is required in order to determine whether the ongoing situation falls into the category of *normal*, *abnormal*, or *unknown* situation.

Additionally, event characterization is a task that depends on the domain knowledge of the considered context. For example, the above example would correspond to a context dedicated to surveillance purposes. It is also a matter of the type of sensor and services deployed in the contexts, since they are likely to constrain the type of events that are going to be generated. However, due to the fact that new services and devices might appear in the context, this circumstance cannot be used as a sole strategy to characterize the different types of event.

A compelling approach for stating the different types of events that are suggestive of abnormal situations consists of using taxonomies. The higher concepts in the taxonomy trees are the ones used as the abstract event types that require further analysis whenever they are matched in the context. Figure 4.3 depicts a brief extract of an OWL taxonomy devoted to characterizing the most characteristic events of a given set of abnormal situations.

This strategy for abnormal situation identification based on the occurrence of characteristic events can be implemented by means of a Multi-Agent System approach. The existence of a *Context Manager* in charge of translating raw sensor and service data into the appropriate lexicon devised for the syntactic dimension of the context model has already been mentioned. Additionally, the *Context Manager* can also be responsible for matching event occurrence that might be suggestive of an abnormal situation. The BDI (Belief-Desire-Intention) model of agency seems to be a compelling approach to cope with the demands involved in dealing with the identification and management of abnormal situations. Therefore, the occurrence of the featured events triggers the goals that have been devised to cope with the abnormal situation. For example, whenever an unauthorized presence event takes place, one of the goals which it triggers is intended to perform an intruder identification for the purpose of eliminating sensor errors. This goal has an associated plan or set of plans for the goal to be achieved. The details concerning the planning strategy will be described later in this thesis.

Part III of this thesis concerns how acting capabilities can be leveraged into Ambient Intelligent systems by implementing a Multi-Agent System approach. However, at this point, it is only relevant that the *Manager* agent is in charge of detecting the occurrence of those events that need to be analyzed with a further level of detail.

The second strategy consists in analyzing the value of those properties that change over time, the so-called *fluent* in the Event Calculus. Given that abnormal situations typically involve changes over time, and considering that fluents model how properties change over time, it is possible to characterize abnormal situation in terms of fluent values and changes. For example, think of an abnormal situation that poses a life threatening hazard whenever the temperature levels increase very quickly to very high values. Most probably, these values are suggestive of an ongoing fire and therefore it requires the system to react in some way intended to extinguish the fire and to prevent people's integrity from being compromised. In any case, temperature sensors have not been devised to alert to quick increases or high temperatures, and therefore it is necessary to implement this functionality at a different level.

Additionally, bear in mind that a quick increase can have different meanings depending on the context. For example, when this temperature increase occurs in a kitchen, it might be due to a cooking activity taking place. Therefore, once again, the fluent characterizations simply outline the possibility of an abnormal situation taking place, and further analysis is also required so as to determine whether or not it is occurring.

The approach followed for characterizing the fluent values and changes suggestive of abnormal situations is based on a commonsensic and qualitative approach. Note that not only fluent values are being considered, but also changes. In this last case, it is not always possible to determine the exact variation quantity, and therefore it is more appropriate to describe changes in terms of qualitative or relative measures. Both strategies, the commonsensic and qualitative approaches, have to be implemented at the knowledge base level, so that additional knowledge can be used to reason about the normality of the situation. At the knowledge-base level, there should be a function intended to evaluate the normality of a fluent value and change. In order to do so, this function should answer the following questions:

1. How does the fluent normally evolve over time?
2. Is the sensed change compliant with the function of time that describes this fluent evolution over time?
3. Is the sensed fluent value still considered normal?

It is a fact that a person cannot be at two different places at the same time (given that those places are geographically independent). In this sense, the fluent that describes a person's position cannot have two different values at the same instant, since it takes time to experience a relevant change in space. Whenever a situation like that takes places, it might suggest that an abnormal situation is taking place.

This strategy is independent of the event taxonomy involved in the event characterization strategy. Given that the implementation of this approach works at the knowledge-base level, it is possible to query the knowledge base about which possible worlds are consistent with the context description at that specific time instant. Those possible worlds are therefore contrasted with the abstract descriptions with which the *Manager* or *Context Manager* agent has been provided. The possible worlds that are consistent with the context description and that at the same time match the situations considered by the Multi-Agent System will cause goals to be triggered and therefore, plans to be instantiated for managing and achieving those goals.

4.7 Interim conclusions

This chapter has been devoted to proposing a comprehensive solution for context modeling and reasoning based on the impact that each level has on composing the final picture of the context. This proposal has been motivated by the conviction that the sole consideration of one of these dimensions does not suffice to capture the complete context picture. Nevertheless, most of the approaches presented to date focus on the semantic layer, using techniques such as ontologies or description logics. Overlooking the syntactic and pragmatic layer leads to poor context models. As result of such a poor models, reasoning tasks are also very limited to the knowledge that can be extracted from the semantic layer. These shortages have motivated the need to propose an approach capable of dealing with the triple dimension of the context modeling.

The ultimate goal of this chapter is to provide a holistic framework to be deployed in Ambient Intelligence, where a bottleneck is found in addressing unexpected situations. For this reason, not only foreseeable situations or scenarios are considered here, but also unexpected ones. The context modeling strategy is especially concerned with how to characterize abnormal situations in such a way that they can be identified and addressed in terms of vague descriptions that are turned into concrete implementations through the planning strategy described in the followings chapters.

Chapter 5

Understanding Context Situations

Chuangtse and Hueitse had strolled onto the bridge over the Hao, when the former observerd, “See how the small fish are darting about! That is the happiness of the fish.” “You are not a fish yourself,” said Hueitse. “How can you know the happiness of the fish?” “And you not being I,” retorted Chuangtse, “how can you know that I do not know?”.

– Chuangtse, c. 300 B.C

Summary – *Given the triple-dimension strategy proposed in the previous chapter for context modeling purposes, this chapter is consequently devoted to describing how such knowledge model can be used to support higher level inferences and understandings of the situations that occur in the supervised context, independently of whether or not they fall within the normality. Special attention is given here to novel situations and how these are to be treated in order to determine whether further responses need to be undertaken to address such situations.*

5.1 Introduction

As a constituent part of human mental events, the ability to foresee, understand, and manage *possible-world* and *multiple-context* semantics is what enables humans to cope with novelty. As mentioned in the previous chapter, the notion of possible worlds is used here to refer to those states of affairs or “worlds” in which, given an event ε, it is true in all the worlds considered possible. For example, a specific presence sensor has detected that there is someone in the kitchen. Therefore, in all the worlds considered possible in that scenario, there is someone in the kitchen. Among all those worlds, it might also be possible to consider that there is someone else in the living room, since the only well known fact is that there is someone in the kitchen. If presence at the living room can be determined, and it can be concluded that there is not no one in there, the world in which there is someone in the living room is no longer considered possible.

Recalling the concept of context, it is here understood as the set of facts or propositional knowledge that describes a specific state of the world, in the same way that J. Allen refers to the concept of *world* in [4]. This concept is represented by a set of descriptions of both the static and dynamic aspects of the world, therefore modeling what is known about the past, present, and future. By using the J. Allen nomenclature, the static aspects of the world are easily captured as *properties* while dynamic aspects are captured by *occurrences* or *events*.

The notion of *multiple contexts* is closely related to that of possible worlds and it refers to the mechanism used to concurrently handle the possible-world semantics at the knowledge-base level.

According to the model proposed for the semantic dimension of the context, the multiple-context mechanism supports action and event modeling by describing the state of the world before, during, and after the action or event took place. Since world changes, or world dynamics, are determined by the events or occurrences that take place in it, it is therefore sensible to model actions and/or events in terms of the world-states that are involved, before, while, and after the event or action takes place. For example, a `person moving` event gives rise to a new world state in which the person that moves has changed his/her location. If a person moves from the kitchen to the living room, the world-state before the event takes place is described by the person being present in the kitchen, while the world state after the event takes place is described by the fact that the person is now located in the living room. Now, think for a moment that the person, before moving, decides to pick up an object; given this scenario, could you correctly answer where the object is after the moving event? Moreover, what if now the object is known to be slippery, what will happen then?

Answering these questions involves holding a great deal of implicit knowledge known as common sense. Ideally, this type of knowledge is contained at the pragmatic dimension of the context model, which not only involves implicit knowledge about how the world works, but also the explicit knowledge about the context domain. Further in this chapter, the difference between implicit and explicit knowledge is analyzed in detail. Meanwhile, for contextualization purposes think of the following scenarios extracted from [100] in which the implicit knowledge held by humans, so-called common-sense knowledge, plays an essential role in understanding the situation:

1. In the kitchen, Lisa picked up the newspaper and walked into the living room.
2. Lisa put a book on a coffee table and left the living room. When she returned, the book was gone.
3. Jamie walks to the kitchen sink, puts the stopper in the drain, turns on the faucet, and leaves the kitchen.
4. Kimberly turns the fan's power switch on.

In the first scenario, it is easily inferred that since Lisa was intially in the kitchen, she picked up the newspaper while she was there and then took it into the living room. It is also obvious to us that if Lisa is in the kitchen she cannot be in any other room at the same time, since we are considering rooms as non-overlapping spaces in a house. With regard to the second scenario, we can easily infer that if Lisa left the living room, she is no longer there, and that if the book is not there when she returns, something must have happened because things tend to remain in the state they are unless a partiuclar event affects them. The "*frame problem*" concerns the determination of those things that can be assumed to stay the same from one moment to another. In the third scenario we easily conclude that, after a while, the water will start spilling onto the floor. Finally, with regard to the question of what will happen in the fourth scenario, we can assume that if everything works as expected, the fan will start up.

These examples make evident the need to involve common-sense knowledge when it comes to interpreting context events. Analyzing in isolation, or even taking into account correlative patterns between context events might help in recognizing certain sets of simple activities. However, when more sophisticated situations take place in the context, which by the way are those that have traditionally required human supervision, only by resorting to implicit knowledge can they be susccessfully addressed, resembling human responses. This chapter is therefore devoted to analyzing all those aspects that have already been stated in the previous chapter. These issues are discussed together here

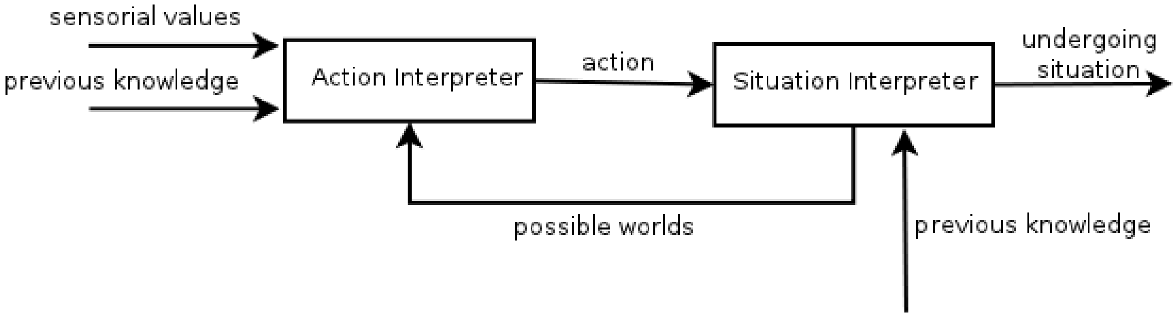


Figure 5.1: Overall view of the process of sensorial information understanding.

in order to show how they help in understanding complex situations from the information provided by the sensorial services deployed in the Ambient Intelligence environment. To this end, the next section justifies the election of Scone as the knowledge-base system used to hold both the implicit and explicit knowledge, and it is particularly useful for managing multiple-contexts and possible-worlds. The following section relates the key issues of common sense, described in section 3.3. Finally, continuing with the first description provided in the previous chapter about how to identify abnormal situations, that section proposes a Multi-Agent System approach for this endeavor. Finally, interim conclusions are drawn in the last section, summarizing the most relevant details of the context understanding strategy described in this chapter.

5.2 Possible-worlds and multiple-contexts semantics

The task of understanding context situations is indubitably related to that of interpreting the information retrieved from the sensorial sources deployed in the supervised environment. Roughly speaking, it can be said that the context dynamics is captured by the environmental sensors, in such a way that context changes are translated into new sensor measurements. The other way round, changes in the values sensed by the environmental sensors can be suggesting that changes are taking place in the context. Considering that context changes are produced by the ongoing actions or events, the task of translating context information into ongoing situations consists in devising the most appropriate set of actions that comply with the sense values and the knowledge that is already known.

Figure 5.1 depicts the overall process involved in mapping context-gathered data onto situations that are likely to be involved in the supervised context. As can be seen in from the same figure, the first stage of the process consists in determining the actions that are occurring in the context, based on a combination of the sensed information and the implicit and explicit knowledge of the context. On the basis of the identified actions, and the previous knowledge, it can be feasible to determine the state of affairs that might be compatible with such information. In other words, provided the identified actions and the domain and implicit knowledge, it is plausible to determine the set of possible worlds that are compliant with such information.

The proposed approach for context understanding therefore consists in modeling actions and situations in terms of the possible worlds before, during, and after they take place. These descriptions are then employed in seeking those actions or situations that match the current state of the context. Those worlds that comply with the context description are considered to be possible worlds. Finally, only those actions or situations that have been described to be compatible with those worlds considered possible are suggested as possible interpretations of the ongoing situation.

Despite the fact that the theory of possible worlds is undeniably associated with agents, yet nothing has been said about the role they played in the theory of possible worlds. Regardless of how agents are implemented, they are expected to hold knowledge about the context dynamics and the previous and implicit knowledge that also describes the context domain. Particularly, the use of possible worlds for context understanding in Ambient Intelligence expects the environment itself to behave as an agent. In this sense, it is the responsibility of he sensorial devices to gather new knowledge that proves the occurrence of actions. Additionally, it is also its responsibility to combine this new knowledge, so-called explicit knowledge, with the previous and implicit knowledge that could lead to more complex inferences. Finally, and more importantly, that previous knowledge contains information about the environment behavior, information regarding the goals, intentions, and beliefs of the agent.

Having said that the agent previous knowledge contains information about the goals and intentions it pursues, it seems obvious that the most suitable approach for implementing this type of agents is that based on a BDI model of agency. Additionally, the previous chapter has already introduced the role of the *Context Manager* in charge of gathering the information provided by the environmental data. Consequently, this agent can easily assume the role of combining such information with the existing knowledge, resulting in a set of worlds that are considered possible for complying with the specifications of the current state of affairs. Finally, those situations that lead to any of those worlds considered possible are suggested as matching the ongoing situation.

5.2.1 Multiple context mechanisms for describing actions and events

Before dealing with the insights of the responsibilities attached to the Context Manager agent, it is necessary to describe how the formal theory of possible worlds is translated into a concrete implementation solution. In this regard, the intention of this subsection is to describe how knowledge should be stated using a possible world approach rather than adopting a propositional perspective, and how, when combined with common-sense knowledge, it can lead to the statement of an *accessible worlds* network. Briefly, this accessible world network supports the identification of those situations that might be compliant with the worlds that are considered possible at each different state of the context.

The first step therefore consists in identifying all the different possible worlds. In order to do so, and given that changes occur as a result of action or event occurrence, the task of identifying possible worlds is therefore approached as a question of describing how actions cause these worlds to evolve from one state to a different one. This approach is inspired in the work of Moore, who in [98] presents a formalization approach for action description based on the theory of possible worlds. Such a formalization approach can be extrapolated and integrated into a different theory, such as the *multiple-context mechanism* proposed by Fahlman in [40].

The multiple-context mechanism is provided as an essential feature of the Scone Knowledge-Base system. As for the possible world theory, the multiple context mechanism allows the representation of different states of affairs, which simultaneously co-occur in the same knowledge-base, without leading to inconsistencies. Note that one of the main objectives with which Scone was conceived was to emulate humans' ability to store and retrieve pieces of knowledge, along with matching and adjusting existing knowledge to similar situations. To this end, the multiple-context mechanism implements an effective means to tackle this objective. The multiple-context mechanism also provides an efficient solution by which to tackle the canonical problem of Artificial Intelligence, which is the *frame problem*. The multiple context mechanism permits the instantiation of new contexts as *virtual copies* of existing ones. Additionally, in any of those contexts it is possible to state those aspects that do not hold as well as the new circumstances that might be relevant for the context description. The fact that a virtual copy behaves as a real copy but does not require such information to be physically replicated provides a very compelling means of addressing the *frame problem*

In any case, the great potential of the multiple-context mechanism used by Scone can be better understood by using the example provided by Fahlman in [40]. The analogy provided uses the Harry Potter novels, in which a fictitious world is presented in which, for example, brooms can fly. This example is especially descriptive because Harry Potter spends time in both worlds, the *wizarding* and the real (or *muggles'*) world. Information about these two worlds must necessarily be kept in the knowledge-base.

Since the "Harry Potter World" (or *wizarding* world) is quite similar to the real world, a new context, "HPW", can be created as an instance of the real world[1]. As has been already mentioned, there are differences between these two contexts, such as the fact that in the "HPW" context a broom is a vehicle. This fact can be easily stated in the "HPW" without affecting real world knowledge, in the same way that knowledge of the real world could be cancelled so as to not be considered in the "HPW" context. The way in which Scone handles multiple contexts so as to avoid incongruence problems is by **activating one context at a time**. By doing this, only the knowledge contained in the active context is considered for the reasoning and inference task.

Unless otherwise stated, the knowledge described in a parent context is inherited by the child context. The context itself is also a node and, like the other the nodes, it stores a set of marker-bits. One of these marker-bits is the context-marker. This bit, when enabled, determines the activation of all the nodes and links that are connected to the active context.

Aside from the role that the multiple-context mechanisms play in supporting the possible world theory, what it is most important is the role it plays in describing actions and events. Representing actions and events in Scone simply consists of defining three new contexts, one describing the world before the action or event takes place and another that represents the state of the world afterward, and one that describes the world properties that hold all along the action performance. In this sense, each of these contexts can be conceived as a possible world, in which the `after` context world is accessible from the `before` context goal when the described action takes place. The following example describes a simplified definition of the `move` event using a syntax similar to that employed by Scone.

```
NEW-EVENT move
  :roles
     origin is a place
     destination is a place
     moving-object is a person
  :throughout
     origin differs from destination
  :before
     moving-object is located in origin
  :after
     moving-object is located in destination
```

In accordance with the aforementioned representation of the `move` event, the propositional knowledge describing the explicit fact of Lisa moving, expressed as `Lisa moves`, can be also presented as an individual instance of the `move` event. This individual instance corresponds to the specific occurrence of Lisa moving from the kitchen to the living room.

---

[1] Using the Scone terminology, "general" is the context node that holds knowledge about the real world, and "HPW" would be an individual node, connected by an is-a link to the "general" node.

The declaration of a new instance of the type `move` event implies that, the new instance named `Lisa moves` inherits the implicit knowledge of the upper type, the `move` event. Since the origin and destination of the `Lisa moves` event have been set respectively to `kitchen` and *living-room*, the Scone Knowledge-Base can be queried about the location of Lisa at two different time instants or at two different *worlds*, one before the action takes place and another, after it takes place. Note how the Knowledge-Base consistency is not affected by that fact that Lisa's location is set to two different places. The use of multiple contexts allows the Knowledge-Base to hold and manage *a priori* inconsistent information in a simple and efficient manner.

```
new—event—indv lisa moves instance—of move
the origin of lisa moves is kitchen
the destination of lisa moves is living—room
the moving—object of lisa moves is Lisa
in—context before
statement—true? lisa is in living—room
  => No
get the location of Lisa
  => kitchen
in—context after
statement—true? Lisa is in living—room
  => Yes
```

The answers provided by the Scone system depend on the context that is active at that moment. In this sense, when the active context is set as the `before` context, the location of Lisa is therefore stated to be the `kitchen`. Whenever the active context changes to the `after` context, the location of Lisa is also changed to the `living-room`.

The most relevant feature of the multiple-context mechanism implemented by Scone is that it supports the construction of a context network along with a context activation scheme. It means that, depending on the desired information, different contexts are activated and deactivated. This feature is particularly important for implementing some of the key issues of common sense that are related to the effects of events.

As has already been mentioned, the effects of events are described in the `after` context of the event description. However, this statement needs to be further developed when the event considered involves other than direct effects, as might be, indirect, canceling, or delayed effects. The following subsections describe the mechanisms that have been implemented in Scone in order to provide support to these sorts of complex effects.

Direct effects of events

In order to illustrate the Scone support strategy implemented for representing direct effects of events, one of the examples presented in [100] is recalled here:

> "*Given that Lisa picked up the newspaper, and this piece of common-sense knowledge, we should be able to infer that Lisa was then holding the newspaper.*"

The previous examples have used a syntax similar but not equal to the one proposed by Scone. However, the following examples are using the Scone syntax. It should be clarified that the description of some events involves the statement of some additional elements, or *roles*, as they are called in Scone. For example, the description of the `take up` event involves several roles, such as the role that represents the object being picked up in the action described, or the place in which the object is being taken up. These roles should be instantiated whenever a new instance of such an event is created.

```
(in-context {general})
(new-event-type {take up} '({event})
                        :ROLES
                        ((:type {pickedObject} {thing})
                         (:type {pickedObjectLocation} {place})
                         (:type {pickerLocation} {place})
                         (:type {picker} {person}))
  :THROUGHOUT
  ((new-eq {picker} {person})
   (new-eq {personlocation} {pickerlocation})
   (the-x-of-y-is-z {personlocation} {picker} {pickerlocation}))
  :BEFORE
  ((in-context (new-context {take up bc}))
   (new-statement {picker} {is located at} {pickedObjectLocation})
   (new-statement {pickedObject} {is located at} {pickedObjectLocation})
   (the-x-of-y-is-z {pickerLocation} {take up} {pickedObjectLocation})
   (new-eq {pickerLocation} {pickedObjectLocation})
   (new-eq {personLocation} {pickerLocation}) )
  :AFTER
  ((in-context (new-context {take up ac}))
   (new-statement {picker} {is located at} {pickerLocation})
   (the-x-of-y-is-z {pickedObjectLocation} {take up} {pickerLocation})
   (new-statement {pickedObject} {is located at} {pickerLocation})
   (new-eq {pickedObjectLocation} {pickerLocation})
  (new-statement {picker} {is holding the} {pickedObject})))
```

The above code listing describes the `take up` event in terms of the possible worlds involved before, after, and during the event occurrence. For example, throughout the action performance, the role assumed by the `picker` is treated equally as a person. Additionally, it can be said about the `before` context that the `picker` is located at the same place as the object that assumes the role of the picked up object. Note that the set of statements that comprise the `before` context description are basically devoted to describe a world in which both the person that is going to pick up the object, and the object that is going to be taken up, are in the same location. In other words, in order for a person to take up an object, the person should be in the proximity of that object, so that s/he can reach it. The description of the `after` context shows a world in which the location of the object picked up is now determined by the location of the person that has taken up the object. Additionally, it can be therefore stated that the person is holding the object.

The following code listing describes how an instance of such an event can be created, and how implicit knowledge is inherited from the upper event type, the `take up` event, leading to important conclusions.

```
CL-USER> (new-event-indv {Lisa takes up} {take up})
{events:Lisa takes up}

CL-USER> (in-context {take up bc})
{events:take up bc}

CL-USER> (the-x-of-y-is-z {pickedObject} {lisa takes up} {lisa
  newspaper})

CL-USER> (the-x-of-y-is-z {picker} {lisa takes up} {lisa})
```

```
CL-USER> (statement-true? {lisa} {is holding the} {lisa newspaper})
NIL

CL-USER> (in-context {take up ac})
{events:take up ac}

CL-USER> (statement-true? {lisa} {is holding the} {lisa newspaper})
{events:picker is holding the pickedObject (0-2655)}
```

The first sentence is intended to create a new instance of the type `take up` event. Additionally, the roles involved in such an event are instantiated so as to assign the `Lisa newspaper` to the role of `pickedOjbect` and the role of `Lisa` to the `picker`. Afterward, still under the `before` context, the Scone system is queried about the truth value of the proposition that states that *Lisa* is holding the *Lisa newspaper*. As can be seen, Scone determines that this propositional statement is false at the `before` context. However, if the active context is changed to the `after` context, then the same query affirms that `Lisa` is holding the `Lisa newspaper`.

Context sensitive effects

A bit more complex to describe are those effects that change upon varying context circumstances. Consider, for instance, the example stated by Mueller in [100]:

> "*We should be able to represent that, if a person picks up a slippery object and is not careful, then the person will not be holding the object.*"

In order to achieve this requirement, the `take up` event needs to be described in such a way that an additional `after` context is described for each of properties that are leading to different effects. For example, one of these susceptible properties is the level of attention paid when holding the object. In this sense, the previous description of the `take up` event needs to be modified in order to include a new context that describes the situation in which the slippery object is held carefully.

```
:AFTER
   ((in-context {take up ac})
   (new-statement {picker} {is located at} {pickerLocation})
   (the-x-of-y-is-z {pickedObjectLocation} {take up} {pickerLocation})
   (new-statement {pickedObject} {is located at} {pickerLocation})
   (new-eq {pickedObjectLocation} {pickerLocation})
   (new-statement {picker} {is holding the} {pickedObject})

  (in-context (new-context {take up slippery object ac} {take up ac}))
   (new-statement {picker} {is located at} {pickerLocation})
   (new-statement {pickedObject} {is located at} {pickerLocation})
   (new-not-statement {pickedObject} {is located at} {pickerLocation})
   (new-statement {pickedObject} {is located at} {pickerLocation})
   (new-not-statement {pickedObject} {is located at} {pickerLocation})
   (new-statement {picker} {is holding the} {pickedObject})
   (new-not-statement {picker} {is holding the} {pickedObject})

   (in-context (new-context {take up slippery object with attention
     ac}  {take up ac}))
   (new-statement {picker} {is located at} {pickerLocation})
   (the-x-of-y-is-z {pickedObjectLocation} {take up} {pickerLocation})
   (new-statement {pickedObject} {is located at} {pickerLocation})
   (new-eq {pickedObjectLocation} {pickerLocation})
```

```
  (new-statement {picker} {is holding the} {pickedObject})))
```

Note how the `take up ac` context describes those aspects of the `take up` event that are satisfied independently of the circumstances that might affect some other effects of the event. Additionally, it is also worth mentioning that it cannot be stated that the `picker` is holding the object in the `after` context due to the fact that attention has not been paid. On the contrary, when circumstances suggest that the `picker` is paying attention to the action, then it can be asserted that the effect of taking up the slippery object with attention is that of the person holding the object.

Creating a new instance of the `take up` event now involves determining the `after` context that should be applied, on the basis of the context sensitive property, which as stated above, is determined by the level of attention. The following code listing describes how to accomplish the selection of the appropriate `after` context, by evaluating the truth value of the statement that describes the level of attention of the `picker`.

```
  CL-USER> (if (is-x-a-y? (get-the-x-of-y-in-context {pickedObject}
                {lisa takes up a slippery object}) {slippery object})
    (if  (statement-true? {lisa} {pays attention to}
         {lisa takes up a slippery object})
         (the-x-of-y-is-z {after context} {lisa takes up a slippery
           object} {take up slippery object with attention ac})
         (the-x-of-y-is-z {after context} {lisa takes up a slippery
           object} {take up slippery object ac}))
         (the-x-of-y-is-z {after context} {lisa takes up a slippery
           object} {take up ac}))

  {events:take up slippery object ac is the after context of Lisa
    takes up a slippery object (0-2696)}
```

Since, at this stage nothing has been said about *Lisa* paying attention to the action of taking up a slippery object, the `after` context that should be inherited is that of taking up a slippery object, without paying attention. On the contrary, something could have been said about the level of attention being paid, as is shown in the following code listing:

```
CL-USER> (new-statement {picker} {pays attention to} {take up})
{events:picker pays attention to take up (0-2695)}

CL-USER> (if (is-x-a-y? (get-the-x-of-y-in-context {pickedObject}
              {lisa takes up a slippery object}) {slippery object})
             (if (statement-true? {picker} {pays attention to} {take up})
                 (the-x-of-y-is-z {after context} {lisa takes up a
                   slippery object} {take up slippery object with
                   attention ac})
                 (the-x-of-y-is-z {after context} {lisa takes up a
                   slippery object} {take up slippery object ac}))
             (the-x-of-y-is-z {after context} {lisa takes up a
               slippery object} {take up ac}))

{events:take up slippery object with attention ac is the after
  context of Lisa takes up a slippery object (0-2697)}
```

The above code is particularly devoted to selecting the appropriate `after` context given the current context circumstances. However, this can be easily generalized so as to be applied to whatever the events that are subject to the value of changing context properties. In any case, the following code listing depicts how the changes suffered by those properties might affect the effects caused by the event.

```
CL-USER> (get-the-x-of-y {after context} {lisa takes up a slippery object})
{events:take up slippery object with attention ac}

CL-USER> (in-context (get-the-x-of-y {after context} {lisa takes up a
  slippery object}))
{events:take up slippery object with attention ac}

CL-USER> (statement-true? {lisa} {is holding the} {wet glass})
{events:picker is holding the pickedObject (0-2644)}
```

In this example, provided that a statement had been asserted about the fact that the `picker` is paying attention to the `take up` event, when the Scone system is queried about the truth value of a propositional fact stating that the `picker` is holding the object being taken up, it concludes that it is true due to the propositional knowledge it returns as a proof of fact.

Nondeterministic effects

Sometimes, the effects of events cannot be determined beforehand for several reasons. For example, in context sensitive effects, whenever the value of those properties that determine the effects are unknown, it is not possible to determine which context to apply. In this sense, Mueller proposes the following:

> "*We should be able to represent that if a person picks up a slippery object, then the person may or may not be holding the object*"

In order to address such a requirement, the previous description of the `take up` action needs now to be enhanced with the general knowledge that affects the situation in which the `take up` event involves a slippery object, yet ignoring the context sensitive circumstances that might determine whether the `picker` is paying or not enough attention.

```
:AFTER
  (
  [...]
   (in-context (new-context {take up slippery object ac} {take up ac}))
   (new-statement {picker} {is located at} {pickerLocation})
   (new-statement {pickedObject} {is located at} {pickerLocation})
   (new-not-statement {pickedObject} {is located at} {pickerLocation})
   (new-statement {pickedObject} {is located at} {pickerLocation})
   (new-not-statement {pickedObject} {is located at} {pickerLocation})
   (new-statement {picker} {is holding the} {pickedObject})
   (new-not-statement {picker} {is holding the} {pickedObject}))
```

There are several aspects that might vary in the world in which a slippery object is taken up from that in which the object is not slippery. As can be noticed from the previous code listing, it cannot be stated beforehand that after the `take up` event, both the `picker` and the `pickedObject` are going to be located in the same place, since it might be possible for the object to fall and end up on the floor.

The way of determining which `after` context to apply is similar to the one described above, although a new after context comes into play to describe the situation in which nothing has yet been said about attention.

```
CL-USER> *CONTEXT*
```

```
{common:general}

CL-USER> (if (is-x-a-y? (get-the-x-of-y-in-context {pickedObject}
              {lisa takes up a slippery object}) {slippery object})
               (if (statement-true? {lisa} {pays attention to}
                    {lisa takes up a slippery object})
                     (the-x-of-y-is-z {after context} {lisa takes up a
                       slippery object} {take up slippery object with
                       attention ac})
                     (the-x-of-y-is-z {after context} {lisa takes up a
                       slippery object} {take up slippery object ac}))
                     (the-x-of-y-is-z {after context} {lisa takes up a
                       slippery object}  {take up ac}))
{events:take up slippery object ac is the after context of Lisa takes
  up a slippery object (0-2693)}

CL-USER>  (in-context (get-the-x-of-y {after context} {lisa takes up a
  slippery object}))
{events:take up slippery object ac}

CL-USER> (statement-true? {lisa} {is holding the} {wet glass})
It cannot be predicted because: {events:Not picker is holding the
  pickedObject (0-2638)} and  {events:picker is holding the
  pickedObject (0-2637)}

CL-USER> (statement-true? {wet glass} {is located at} {living room})
It cannot be predicted because: {events:Not pickedObject is located at
  pickerLocation (0-2636)} and {events:pickedObject is located at
  pickerLocation (0-2635)}
```

The previous examples show the mechanism that has been built on Scone in order to provide support for describing and managing the nondeterministic effects of events. As can be seen from the previous code listing, the fact that the same statement is said to be true and false leads the system to conclude that the truth value of such a statement cannot be determined given the knowledge held by the system at that moment.

Effects of concurrent events

So far, the description of the effects of an event has been only concerned with describing events as occurring in isolation. However, this is not a realistic vision, and in fact, events need to be described considering the collateral occurrence of events. The majority of the events that are concurrently taking place do not have an impact on each other, although there are some other situations in which the effects of a certain event might be affected by the concurrent occurrence of a different event. In this sense, Mueller states the following:

> “*We should be able to represent that certain concurrent events are impossible; for example, a person cannot walk into two rooms simultaneously.*”

Firstly, it is necessary to describe a new event type, the `walk into` event, which derives from the `walk` and `walk to` events, which for the sake of concreteness have not been listed here.

```
(in-context {general})
(new-event-type {walk into} '({action} {walk} {walk to})
                :roles
                ((:type {enteringRoom} {room}))
```

```
 :THROUGHOUT
 ((new-is-a {enteringRoom} {enclosed space}))
 :BEFORE
   ((in-context (new-context {walk into bc}))
    (new-statement {walker} {is located at} {from})
    (new-statement {from} {is connected to} {enteringRoom})
    (the-x-of-y-is-z {walkerLocation} {walk into} {from}))
 :AFTER
   ((in-context (new-context {walk into ac}))
    (new-statement {walker} {crosses across}
      (get-the-x-role-of-y {doorway} {enteringRoom}))
    (new-statement {walker} {is in} {enteringRoom})
    (new-statement {walker} {is located at} {enteringRoom})
    (the-x-of-y-is-z {walkerLocation} {walk into} {enteringRoom}))))
```

The reason that a person cannot enter two different rooms at the same time is a direct consequence of the physical space property and the space and position that objects occupy in that physical space. In this sense, the following code listing describes how both rooms, the `kitchen` and the `living room` are stated as two different rooms that cannot be considered equal.

```
(in-context {general})
(new-indv {kitchen} {room})
(new-indv {living room} {room})
(new-not-eq {living room} {kitchen})

(new-event-indv {Lisa walks into} {walk into})

(in-context {walk into bc})
(the-x-of-y-is-z {from} {Lisa walks into} {bedroom})

(in-context {walk into ac})
(the-x-of-y-is-z {enteringRoom} {Lisa walks into} {kitchen})
```

Provided that both rooms are different, the following code listing describes the event of Lisa entering a room. Note that this example is aimed at describing how concurrent events can be addressed in Scone. For that reason, the examples shown here overlook some relevant aspects such as those involving time. Nevertheless, those aspects will be described later on, in the following section.

```
CL-USER> (in-context {walk into ac})
{events:walk into ac}

CL-USER>(statement-true? {lisa} {is located at} {kitchen})
{events:walker is located at enteringRoom (0-1946)}

CL-USER> (statement-true? {lisa} {is located at} {living room})
NIL

CL-USER>(statement-true? {lisa} {is located at} {bedroom})
NIL

CL-USER> (the-x-of-y-is-z {enteringRoom} {lisa walks into} {living room})
{events:living room} cannot be the {events:enteringRoom} of
{events:Lisa walks into}.  Continuing...
NIL
```

Having said that the previous sentences are considered to take place at the same time instant, the last sentence shows how the Scone system rejects the propositional statement that would make Lisa enter both rooms at the same time. In this sense, the last sentence is intended to assert the fact that Lisa is walking into the living room. However, Scone fails to assert this statement because previously, the `enteringRoom` role of the `Lisa walks into` event has been set to `kitchen`. Since `kitchen` and `living room` are not equivalent, and since a person cannot enter two different locations at a time, Scone returns a message notifying of this failure attempt.

If both rooms would have been described as equivalent rooms, for example, a room working as a kitchen and living room at the same time, it is possible for a person to enter both rooms at the same time, because these two rooms are spatially equivalent.

Besides those effects of events that cannot take place concurrently, there are some other types of effect of events that, when globally considered, produce a different result from the one that would be expected if the event had been considered individually or in isolation. One of these types of effects is the cumulative or canceling effects of events. In this sense, Mueller states that:

> "*We must be able to reason about concurrent events with cumulative or canceling effects. For example, if a shopping cart is pushed, it moves forward. If it is pulled, it moves backward. But if it is simultaneously pulled and pushed, then it moves neither forward nor backward; instead, it spins around.*"

In this sense, in order to determine whenever the effects of a certain event are being canceled out, it is necessary to state in the first place, which other events are capable of producing canceling effects to the first one. In this sense, a new relation type has been defined in order to enumerate the other events that are producing canceling events. The "`cancel effects of`" relationship is a transitive relation intended to establish those pairs of actions that are producing canceling effects. Additionally, a new function needs to be defined in order not only to determine whenever canceling or cumulative effects might take place, but also to be responsible for devising what the new `after` context would be as a result of these effects of concurrent events taking place. The following function is therefore intended to combine and produce the after context resulting from several concurrent events taking place.

```
(defun get-the-after-context-of-concurrent-event (x y z)
  (setq x_ac (get-the-x-of-y {after context} x))
  (if (not (is-x-a-y? x {compound event}))
      (format t "~A␣should␣be␣a␣compound␣event" x)
      (if (statement-true? y {cancels effects of} z)
          (progn
            (format t "~A␣cancels␣the␣effects␣of␣~A" y z)
            (setq current_context *context*)
            (in-context x_ac)
            (new-statement y {cancels effects of} z)
            (in-context current_context))
          (progn
          (with-temp-markers (m m1)
              (loop for i in (list-context-contents
              (get-the-x-of-y {after context} y))
                do(progn
                     (mark-context-contents
                     (get-the-x-of-y {after context} y) m)
                     (mark-context-contents
                     (get-the-x-of-y {after context} z) m1)
```

```
                      (loop for j in (list-marked m)
                         do (connect-wire :context (lookup-element j)
                         (lookup-element x_ac)))
                      (loop for j in (list-marked m1)
                         do (connect-wire :context (lookup-element j)
                         (lookup-element x_ac))))))))))
```

The following code listing describes how such functions can be applied to an instance of an event of type`push` and `pull`, such as `pullandpush` event, that occur concurrently.

```
CL-USER> (get-the-after-context-of-concurrent-event  {pullandpush} {push} {pull})
{push} cancels the effects of {pull}
{common:general}

CL-USER> (list-context-contents {pullandpush ac})
({events:push cancels effects of pull (0-1996)})
```

As can be seen from the previous code listing, a new `after` context has been created so as to include the effects of the two events, `push` and `pull`, when concurrently performed. In this particular case, it states that the `push` effects are canceled by the `pull` ones.

Indirect effects

There are some other types of events whose effects might also affect third party objects, in an indirect manner. In order to illustrate this aspect, Mueller recalls the same example used for the direct effects of events, that of Lisa taking up an object. However, now the focus is on proving that, as an indirect effect of the `take up` event, the object picked up changes its location along with the picker.

> *"Where did the newspaper end up? It ended up in the kitchen" We know that, if a person is holding an object, then the object moves along with the person.*

Scone has been enhanced with several functions in order to address the indirect effects of events. One of these functions is the one intended to give a unique value to a property with changing values, which additionally can only hold one value at a time. For example, as mentioned above, the location property of an object is unique, and for that reason, assigning it a new value causes the previous one to be overwritten. In any case, the important aspect here is the fact that just by changing the picker location property, it causes the picked-up object, that is Lisa's newspaper, to also change its location, due to an indirect effect of Lisa holding that object.

```
CL-USER> (in-context {take up bc})
{events:take up bc}
CL-USER> (get-the-x-of-y-in-context {pickedObjectLocation} {Lisa takes up})
NIL
CL-USER> (get-the-x-of-y {pickedObjectLocation} {Lisa takes up})
{events:living room}
CL-USER> (in-context {take up ac})
{events:take up ac}
CL-USER> (get-the-x-of-y {pickedObjectLocation} {Lisa takes up})
{events:living room}
CL-USER> (the-only-x-of-y-is-z {pickerLocation} {Lisa takes up} {Bedroom})
{events:bedroom is the pickerLocation of Lisa takes up (0-1901)}
CL-USER> (get-the-x-of-y {pickedObjectLocation} {Lisa takes up})
{events:bedroom}
CL-USER> (get-the-x-of-y {pickedObject} {Lisa takes up})
{events:Lisa newspaper}
```

Delayed effects and continuous change

Despite the fact that the description of the previous type of effects overlooks the role played by time, the description of the delayed effects and the continuous change can no longer obviate this issue. In this sense, the Scone approach for modeling and describing events needs to be enhanced with the required capabilities to state how the effects of some events might take place over a time interval, or at a specific time instant. Mueller proposes the following descriptive example:

> “*Jamie walks to the kitchen sink, puts the stopper in the drain, turns on the faucet, and leaves the kitchen*”.

In order to address this requirement, several functions proposed for the Event Calculus theory need to be implemented. Recall that the notion of *fluent* is used to describe those properties that change along time. Therefore, in order to determine the value of a fluent property, e.g. the height of the water level, it is necessary to describe how that fluent evolves with time, and at what specific time instant the property needs to be known. The following code listing describes the functions that have been provided to this end:

```
(new-type {fluent} {thing})

(new-relation {holds at}
              :a-inst-of {fluent}
              :b-inst-of {time point})

(new-relation {releases at}
              :a-inst-of {fluent}
              :b-inst-of {time point})

(defmacro get-element-fluent (e fluent &optional (time-point *current-time-point
    *))
  (if time-point
      (setq fluent (read-from-string (concatenate ’string
      (write-to-string fluent) "-at-"  (remove #\}
      (subseq  (write-to-string time-point) 1))))))
      ‘(getf (properties ,e) ,fluent))

(defmacro eval-element-fluent (e fluent)
  ‘(eval (getf (properties ,e) ,fluent)))
```

These functions form an extract of the event calculus functions that have been implemented. Upon the description of the `turn on faucet` event, several conclusions can be reached regarding the changing value of the water level property.

```
(NEW-EVENT-TYPE {turn on faucet} ’({event} {turn on} {turn off faucet})
   :ROLES
   ((:indv {turnedOnFaucet} {faucet}))
   :BEFORE
   ((in-context (new-context {turn on faucet bc}))
    (new-statement {turnedOnFaucet} {status} {off}))
   :AFTER
   ((if(eq (get-the-x-of-y {sinkTapStatus} {faucetSink}) {opened tap})
       (progn
         (in-context (new-context {turn on faucet ac}))
         (new-statement {turnedOnFaucet} {status} {on})
```

```
          (new-statement {faucetLiquid} {is being dropped through}
          (get-the-x-of-y {sinkPipe} {faucetSink})))
        (progn
          (in-context (new-context {turn on faucet ac}))
          (new-statement {faucetLiquid} {is contained in} {faucetSink})
          (in-context (new-context {turn on faucet DE}))
          (new-statement {faucetSink} {gets overflowed with} {faucetLiquid})
          (new-statement {faucetSink} {is dropping} {faucetLiquid})))))
```

The following code listing shows how Scone can be enhanced with event calculus functions that allow it to reason about the delayed effects of events and continuous change.

```
CL-USER> (in-context {turn on faucet bc})
{events:turn on faucet bc}

CL-USER> (list-context-contents *context*)
({events:LEVEL of faucetLiquid holds at instant T0 (0-2586)}
[...])

CL-USER> (in-context {turn on faucet ac})
{events:turn on faucet ac}

CL-USER> (list-context-contents *context*)
({events:LEVEL of faucetLiquid holds at instant T2 (0-2590)}
 {events:LEVEL of faucetLiquid releases at instant T1 (0-2589)}
 {events:faucetValve status on (0-2588)})
```

The activation of the `before` context describes a world in which, at a time instant labeled as `T0`, the water level does not change, since the tap has not yet been opened. Changing the active context to the `after` context, shows a world in which the water level will ultimately hold at time instant `T2`, however, it will be released from the common-sense law of inertia at time instant `T1`. It means that, at `T1` the water level changes its value according to the function of time with which this fluent has been described. The following code listing shows an example of such a function of time for the water level case.

```
CL-USER> (properties (lookup-element {faucetliquid}))
(:LEVEL-AT-INSTANT-T2 FULL :LEVEL-AT-INSTANT-T1 (* FLOW (/ T_S BASEAREA))
 :LEVEL-AT-INSTANT-T0 EMPTY :LEVEL FULL :ENGLISH-NAMES
 (("faucetLiquid" . :INVERSE-ROLE)))

CL-USER> *current-time-point*
{events:instant T1}

CL-USER> (get-element-fluent  (lookup-element {faucetliquid}) :level)
(* FLOW (/ T_S BASEAREA))

CL-USER> (get-element-fluent  (lookup-element {faucetliquid}) :level {instant-t0
    })
EMPTY

CL-USER> (get-element-fluent  (lookup-element {faucetliquid}) :level {instant-t2
    })
FULL

CL-USER> (get-element-fluent  (lookup-element {faucetliquid}) :level {instant-t1
    })
```

```
(* FLOW (/ T_S BASEAREA))
```

As can be seen from the above code listing, the water level fluent changes its value depending on the time instant considered.

5.3 Description of the context understanding process

Figure 5.1 succinctly describes the overall process of understanding the activities that are being carried out in the context on the basis of the information retrieved from the environmental sensors and services. Basically, events take place in the context simultaneously causing the sensorial devices to notice changes. These changes result in new sensor values being measured and captured by the *Context Manager* agent. Recall that this agent is in charge of supervising all the information that is generated in the context by the sensor mechanisms deployed in it.

Section 4.6 introduces the most relevant aspects of an approach intended to identify the occurrence of abnormal situations. In essence, the identification of abnormal situations is the first requirement that should be satisfied by a system intended to support Ambient Intelligence. Aside from the pre-coded reactions to foreseen situations, only when a situation has been considered abnormal is the system entitled to take some decisions intended to comply with the goals and intentions that have been enacted for that context. The decision about how to respond to that situation is also affected by the mechanisms that are available to implement the response. Nevertheless, these aspects will be addressed in Part III, as part of the system capability to implement behavioral capabilities.

This section therefore focuses on thoroughly describing the foundations of the situation characterization process. Such a process is mainly grounded in the distinction established between the different types of effects of events, described in the previous section. In this sense, the proposed approach is not only concerned with which events, in a direct manner, could be responsible for the captured effects, but also, these effects have to be analyzed from the perspective of the more complex effects caused by event occurrences, such as those involving delayed or canceling events, among others.

Although a description of the characterization process is provided at the end of this section, for the sake of clarity, a case scenario is provided so as to illustrate the characterization process of some of the situations in which other than direct effects of events are involved.

5.3.1 A case scenario describing the understanding process

The following lines describe the events gathered by the `Context Manager` agent as a result of several sensors noticing the changes that are taking place in the context. At this point, it can be stated that the use of the word *event* has a twofold dimension. On the one hand, it refers to the general description of events in the sense of actions. On the other hand, it refers to a specific occurrence, and in this sense it is therefore assimilated to the effects detected by a sensor device. Throughout this section, the following lines are indistinctly referred to as events or effects of events, due to the twofold dimension already mentioned.

```
2011-05-02 00:59 presence-sensor-at-kitchen activated
2011-05-02 00:59 kitchen-lamp turned-on
2011-05-02 00:59 presence-sensor-at-bedroom activated
```

Whereas these events comprise the dynamic information of the context, so-call the domain knowledge, has already been asserted in the Scone Knowledge-Base. Static information comprises several statements such as those describing that the above events take place in a house in which a single person lives. Let's say that Lisa is the person that lives in that house, and that she is an elderly woman with some mobility problems that do not allow her to move too quickly round the house. Additionally, this is a two-floor house in which the bedroom is located on the second floor and the kitchen on the first floor.

Given the captured events and the information about the context domain, the characterization process is, at first, engaged in determining which set of events ε can lead the previous context situation, labeled as $\mathcal{S}_0$, to the situation in which presence is detected at both, the kitchen and the bedroom, and the kitchen lamp is turned on, also labeled as situation $\mathcal{S}_1$. In this sense, the purpose of the characterization process is to determine the unknown value ε, which refers to a set of events which, when applied to situation $\mathcal{S}_0$, result in the new situation or possible world equal or equivalent to $\mathcal{S}_1$:

$$effectOf(\varepsilon, \mathcal{S}_0) = \mathcal{S}_1 \tag{5.1}$$

The method proposed for finding the value of ε consists in iteratively analyzing whether any of the different types of effects of events could explain the values gathered from the environmental sensors. The understanding process therefore consists in seeking for those events that respectively comply with both situations, $\mathcal{S}_0$ and $\mathcal{S}_1$, in their `before` and `after` context.

The first type of effects to be considered are the *direct effects of events*, with the purpose of determining whether there exists a single event that can connect the situation previously described as $\mathcal{S}_0$ to the one described in $\mathcal{S}_1$. In order to do so, Scone is queried about the existence of an event that complies with a `before` context equivalent to that represented by $\mathcal{S}_0$ and an `after` context equivalent to that represented by $\mathcal{S}_1$. Recall that situation $\mathcal{S}_0$ holds the following propositional knowledge:

1. Lisa is a person
2. Lisa’s house is a house
3. Lisa inhabits Lisa’s house

The situation $\mathcal{S}_1$ holds both the propositional knowledge that describe situation $\mathcal{S}_0$, and the following statements:

1. A moving object is detected in the kitchen at time instant 2011-05-02 00:59
2. The kitchen lamp is turned on in time instant 2011-05-02 00:59
3. A moving object is detected at the bedroom at time instant 2011-05-02 00:59

The first step is therefore intended to seek for an event that matches both situations with its `before` and `after` context, and for that purpose the following function has been devised:

```
CL-USER> list-events(
        :BEFORE
          '(({Lisa} {is-a} {person}) ({Lisa house} {is-a} {house})
          ({Lisa} {habits in} {Lisa house}))
        :AFTER
          '(({movingObject} {is detected at} {kitchen} {2011-05-02 00:59})
          ({kitchen lamp} {status} {on} {2011-05-02 00:59})
          ({movingObject} {is detected at} {bedroom} {2011-05-02 00:59})))
```

The specificity of the propositional statements composing both worlds ensures that no event satisfies the matching process. It is not surprising since the example has been purposely selected so as to show the potential of an approach that considers effects of events other than just direct ones.

The second stage of the characterization process consists in determining whether a concurrent effect of events is being involved in the situation described by both $\mathcal{S}_0$ and $\mathcal{S}_1$.

Due to the fact that considering the propositional knowledge of situations $\mathcal{S}_0$ and $\mathcal{S}_1$ as a whole has been unfruitful, the following approach consists in analyzing those effects that could individually cause those events.

```
CL-USER> list-events(
        :AFTER
         '(({movingObject} {is detected at} {kitchen})))
```

As it can be seen, the time stamp has been ignored since, at this stage, the focus is in determining which events could cause an effect that is equivalent to the one just captured in the kitchen. The result provided by Scone enumerates a list of events that have such a statement in their `after` context. Those events are explored one by one, however, for the sake of concreteness, here, only the `walk into` event is described. In this sense, each of the resultant events is being activated and instantiated in order to evaluate whether they are or not plausible event occurrences. This means that the `walk into` event is instantiated by matching the roles defined for that event to the available individuals instantiated for that specific context, which are `Lisa`, `Lisa's house`, or `kitchen lamp` among others. The instantiation process results in the following individual event being proposed:

```
CL-USER> (new-event-indv {Lisa walks into} {walk into} {2011−05−02 00:59})
{events:Lisa walks into}

CL-USER> (in-context {walk into ac})

CL-USER> (the-x-of-y-is-z {enteringRoom} {Lisa walks into} {kitchen})
```

Please, notice that nothing can be yet said about the location from which Lisa came, since there is not any individual or statement that could lead to a match with the `from` role of the `walk into` event. The second event goes through the same process and leads to the statement of the same individual event, although matching the `enteringRoom` role to the `bedroom` rather than to the `kitchen` as happened with the previous event.

```
CL-USER> (new-event-indv {lisa walks into} {walk into} {2011−05−02 00:59})
{events:Lisa walks into}

CL-USER> (in-context {walk into ac})

CL-USER> (the-x-of-y-is-z {enteringRoom} {Lisa walks into} {bedroom})
```

Both individual events are two different instances of the same `walk into` event type that happen to occur at the same time instant. This example slightly differs from the one proposed when describing the impossible effects of events, in which, the same individual event role, the `enteringRoom`, was assigned to two different and nonequivalent values. On the contrary, the current case scenario presents the existence of two different individual events which, at least *a priory*, do not seem to lead to any error or conflict. However, it seems obvious that Lisa cannot be in the kitchen and in the bedroom at the same time instant. Note that only Lisa can assume the `movingObject role` since it is the only individual instance that can be equivalent to it.

Provided that both events can be *a priori* valid statements, there should be a means to control this sort of situation. In this sense, and as has already been mentioned in the previous chapter, this situation needs to be controlled from the point of view of common-sense knowledge about how certain fluents evolve through time. In this specific case, a person's location is a fluent, since it is a property that changes with time. The proposed approach for the determination of whether impossible effects of events are involved is to convert the question into whether or not the fluents involved in the analyzed events are plausible.

The two instances of the `walk into` event which respectively state that Lisa walks into both the kitchen and the bedroom at the same time instant, are therefore evaluated from the perspective of the fluents that are involved in both events. The only fluent involved is the one concerning the person's position. In this sense, at time instant t_0 Lisa's position is at the kitchen, whereas at time instant t_1 Lisa is at the bedroom. Using the space utility implemented according to the OpenLS standard, the distance in meters between the kitchen and the bedroom can be calculated, and it seems that there is a distance of 200 meters from the kitchen to the bedroom. The knowledge about Lisa that has been asserted to the knowledge-base also includes some information about the speed at which she moves, since, as has already been mentioned she has some disabilities that makes her move slowly. In this case, Lisa is known to move at a speed of 1 km/h. Given that Lisa is in the kitchen at time instant t_0, she cannot be at 200 meter distant 0 seconds later, at time instant t_1, according to the function that describes how the fluent `person location` evolves over time according to her/his speed of movement.

After having explored the situation from the perspective of the impossible effects of events, it can be concluded that the previous individual events cannot be asserted by the knowledge base due to the fact that they comprise two impossible effects of events. This failure means that the role matching process is not correct for the `movingObject` since it has lead to an impossible value of one of its fluents, the `movingObjectLocation`. The next step is therefore intended to seek an additional instance of a `movingObject` type that can assume the role in the `walk into` event instance. If there had been any additional instances of the `movingObject` type, such as for instance an animal or an additional person in the house, the same process would be carried out in order to determine the plausibility of the event instances in which they would enter the kitchen or the bedroom. However, there is not any additional instance of a moving object that can adopt the role in any of the two individual events. The approach followed then is to interpret these types of situation as though a moving object were present in the house. Therefore, a generic instance of a moving object is asserted by the Scone Knowledge-Base.

This changes the whole picture of situation t_1, which is now faced from the perspective of a new statement, as is the fact that a new moving object is present in the house. The assertion of a new statement causes the whole process to start from the beginning, in the light of the new information provided by the recently asserted fact.

Overlooking some aspects that do not lead to any relevant information, the characterization process reaches the point of determining which events are responsible for causing an `after` context in which a new moving object is present in the house. The `incoming guest` or the `overnight guest` events are two examples of events that cause the `after` context to assert that a person is present in the house. Now, the evaluation of the plausibility of these events does not bring to light any incompatibility, and therefore, a different context is created so as to include the possibility of these events taking place. Each of these contexts is analyzed individually, through the same characterization process described in this section.

Following the same dynamic, the different effects of events are analyzed in order to determine whether they are or not involved in the situation being characterized. The last of the effects that is going to shed some additional light on the analysis are the context-sensitive effects of events. It seems obvious that it is not usual to receive a guest at past midnight, but how is that common-sense knowledge articulated in this characterization process? It would not be sensible to state that all guests arrive before midnight because it is not always true, so this fact needs to be stated as a *typical* feature of the event. As has already been mentioned, Scone provides an excellent support for dealing with `typical` knowledge, that is, the type of common-sense knowledge that tends to be true but cannot be categorically affirmed since exceptions are possible. Making the most of the exception handling support by Scone, it is possible to assert information about the typical case without an additional cost, since it can be canceled by specific instances. In this sense, it can be asserted that the typical `incoming guest` does arrive before midnight.

This can be stated by resorting to the capability implemented in Scone to handle context-sensitive effects. An `after` context is defined to state that if it is past midnight, it is not true that a person enters the house. Otherwise, the `after` context that is activated is that resulting in a new statement asserting that a person enters the house.

Due to the fact that it is past midnight the `after` context that applies is the one that states that in the typical case, a person does not enter the house. The fact that this event does not agree with the fact that there is a moving object in the house leads the process to abandon this path and explore other possibilities, such as the fact that a person is staying overnight. However, none of the explored paths leads to a plausible justification of how a moving object is present in the house.

This example shows how after having explored all the possible worlds, the characterization process is not capable of determining the set of events ε that drives $\mathcal{S}_0$ to reach $\mathcal{S}_1$. In this cases, the situation is labeled as `abnormal` and the responsibility is delegated to the *Handler Agent*, which should adopt the most appropriate response to comply with the goals and intentions described for the considered context. These aspects are addressed in the following chapters. However, what concerns the *Manager Agent* level is the responsibility of labeling the situation using one of the categories proposed for the different types of abnormal situations; refer Figure 5.2 in which an extract of the taxonomy is depicted.

5.4 Interim conclusions

Understanding the activities carried out in a context is a task that requires interpreting the effects of events captured from the sensors and services deployed in the context.

In the simplest case, these effects are a direct consequence of an event occurrence, and therefore the understanding process basically consists in identifying all those effects that have been described to cause the noted effect. However, the most common case involves more complex effects of events, and therefore a more profound analysis is required to determine the set of events that have lead to a current situation.

An essential feature of the proposed approach is its foundation in the theory of possible worlds, and how it has been implemented by resorting to the multiple-context mechanism provided by Scone. With some enhanced functions inspired in the possible world theory and a thorough description of the different types of effects of events it is possible to map effects onto causing events as a mean to provide a plausible explanation of the situation that is taking place in the context.

It has to be remarked that unforeseen situations can also be addressed by means of this approach. In these situations, even when a plausible explanation cannot be provided, the system is capable of understanding that an abnormal situation is taking place.

This chapter ends Part II which has been devoted to describing how intelligent systems can be enhanced with cognitive capabilities, grounded in solid common-sense knowledge. The following part is intended to propose a solution for enabling behavioral capabilities in an Ambient Intelligent system, which allows it to respond under those situations that are deflecting the context from the

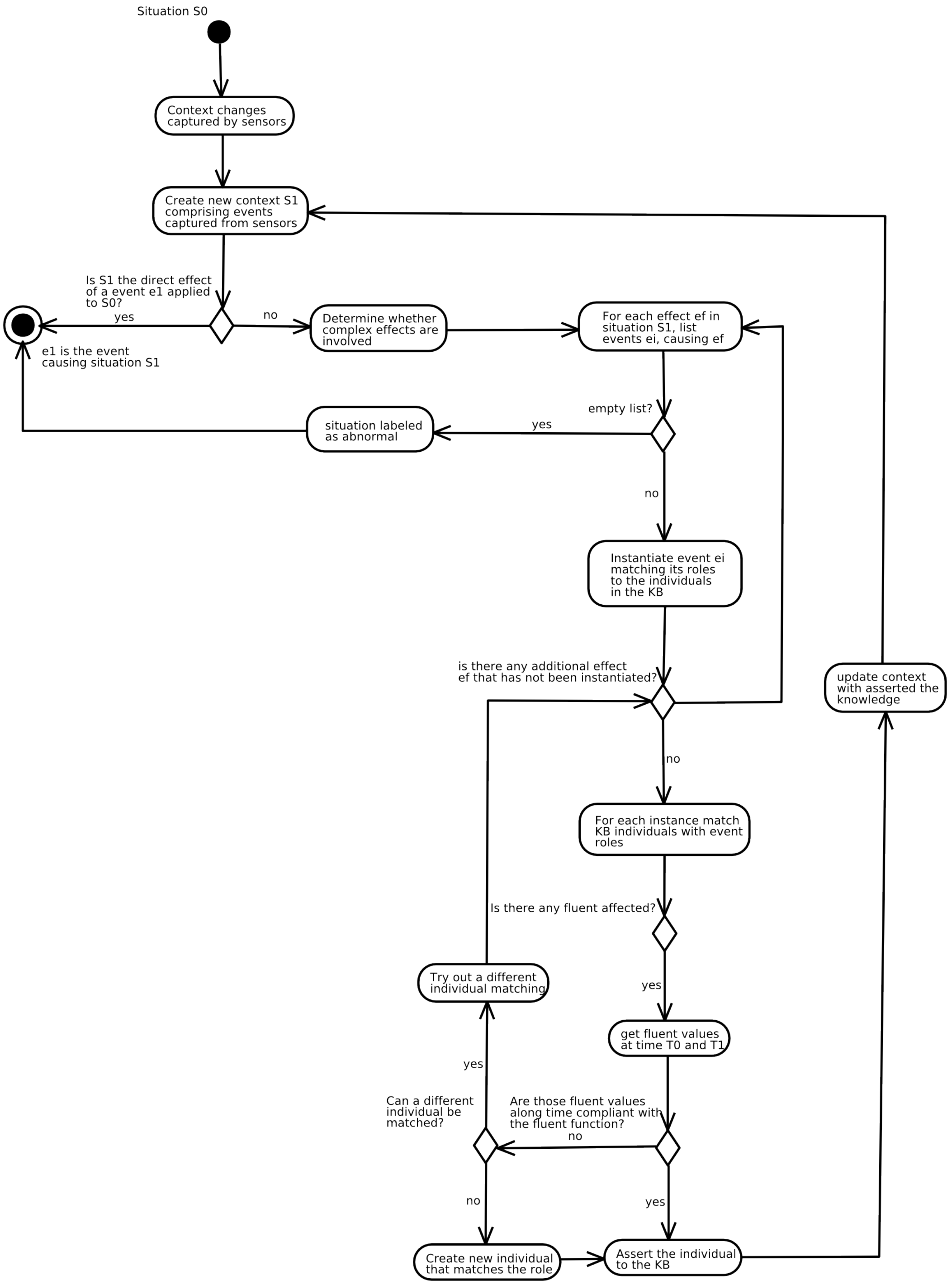


Figure 5.2: Process diagram that outlines how the understanding process is carried out.

desired goal state.

Part III

Acting

Chapter 6

Behavioral Response Generation

"Human behavior flows from three main sources desire, emotion, and knowledge."
–Plato

Summary – *This chapter begins the part dedicated to exploring how Ambient Intelligence systems can be endowed with "acting" capabilities. The previous part has addressed the challenges involved in understanding the context, whereas this part describes the methodological approach proposed for devising when and how Ambient Intelligence systems are expected to respond. Essentially, this chapter is about the role that planning theory plays in the behavioral response generation and the additional issues involved when planning under the Ambient Intelligence scope.*

6.1 Introduction

Part II of this thesis was mainly devoted to describing and analyzing the understanding tasks required as a primary need for the decision making process. These tasks were aimed at constructing feasible explanations accounting for the data gathered from the surrounding context. In this sense, the proposed approach consists in collecting data from the environmental sensors, services, and devices and analyzing them seeking plausible actions or events that might explain them. An action explains a situation when the requirements for the action to take place and the effects that it produces match the context situation before and after the change. Additionally, the identified actions can also be framed in a more general action, here referred to as *ongoing situations*.

In this regard, activities can be seen as plans motivated by a primary reason or a goal. The theory of plans therefore plays an essential role in both facets of Ambient Intelligence systems, which are the cognitive and the reactive facets. The close connection established here between Ambient Intelligence and planning strategies is one of the cornerstones of this thesis.

It is not until now that the role of planning theory has been brought to light, but it does not mean that it has not been used so far. On the contrary, previous chapters have been dealing with planning concepts, although for the sake of clarity of exposition, they have been doing so without stating it explicitly. The reason behind that decision is that the methodological approach previously proposed involves many different theories, among which planning does not play an essential role compared with the rest[1]. However, when it comes to deciding whether a system response is needed and how it is going to be undertaken, the planning theory becomes essential. Plans are therefore involved in both of

---

[1] Recall that the context understating challenges set out in the previous chapters have been addressed resorting to philosophical theories such as that of possible worlds, multiple context, or model of agency, among the most relevant ones.

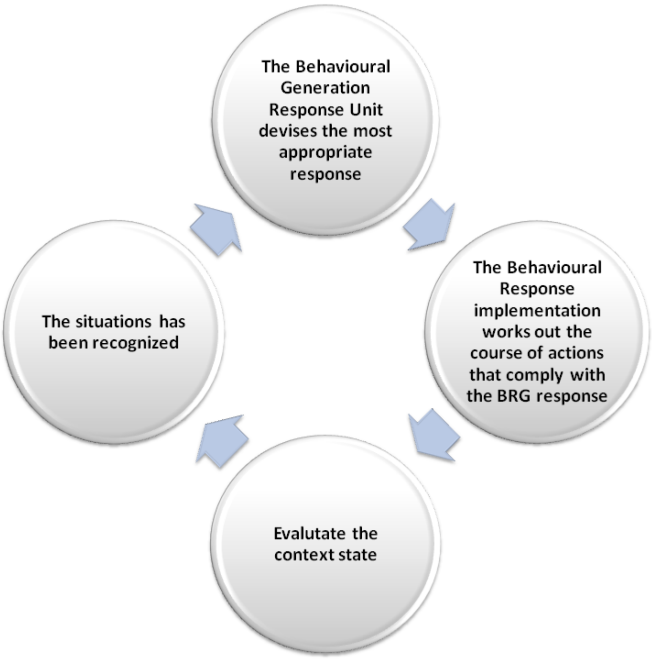


Figure 6.1: Acting process stages

the facets of Ambient Intelligence systems, in both the understanding and acting role. Understanding uses plans to devise the final goal to be pursued, whereas the acting role uses them to devise the course of actions that can lead the system to a desired state, and to determining how those actions are going to be implemented by means of the available services. Figure 6.1 provides an overall and simplified view of the iterative stages involved in this facet of Ambient Intelligence systems, particularly devoted to reacting to context changes.

The action planning theory also underlies the service composition process. More specifically, the focus of this thesis is on achieving automatic service composition as the means of providing articulated responses.

However, despite the essential role that planning theory plays in enabling Ambient Intelligence, it is a fact that the literature in this regard is scarce. This shortage is mainly due to the nonlinearity of problems that involve the exploration of a huge number of states, but also due to some other features such as the nondeterministic effects of events, which makes it impossible to determine whether picking up a slippery object will end in the object being dropped; or delayed effects, that occur some while after the event took place, leading people to foresee that if the kitchen sink has a stopper in, opening the tap will cause the water to overflow. These are some of the features that make planning in Ambient Intelligence a non-trivial issue.

The need for planning strategies in Ambient Intelligence is not a new concern; on the contrary it has already been pointed out in [6]. That paper pays special attention to device heterogeneity, so characteristic of Ambient Intelligence contexts, advocating a distributed-centralized HTN-like approach (Hierarchical Task Network) [38]. In spite of agreeing on the need to address the device dynamism and heterogeneity, here it is believed that these aspects are to be tackled from the middleware perspective, rather than from the planning one. Therefore, device heterogeneity should remain a transparent matter for the planner, as justified in the next section. In this regard, the use of agent approaches is also commonly accepted for auxiliary tasks. The work in [49] highlights the role assigned to a Multi-Agent System (MAS) architecture, acting as the context observer and regulator. The MAS assumes responsibility for providing the planner with the required information about the context and the mechanisms to respond to it.

Although not directly applied but easily extrapolated to Ambient Intelligence, the use of probabilistic search techniques [140] is presented in [66]. This paper addresses the problem of task planning and action selection by means of a fuzzy-neural network approach combined with agent coordination and cooperation methods. Agents are trained to select the most appropriate action depending on the field configuration, changing their selections whenever the objects in the field adopt a different configuration. Another interesting approach, with applications in the Ambient Intelligence field, is the work proposed in [69]. This approach adopts an HTN planning strategy enhanced to fill the gap between real world environments and the planning scenarios. The Script-Based Task Planner (STP) employs a script structure to adapt the planning scenario to the real world environment. Finally, the work in [68] is also relevant. This work presents and discusses a planning strategy that seeks the optimal actions in partially observable stochastic domains, providing a firm foundation for planning under uncertainty conditions of actions and observations. An overall view of the planning strategies, under distributed and cooperative circumstances, is presented in [106]. The conclusions and suggestions drawn from this study set the basis for identifying the shortcomings of traditional Artificial Intelligence planning strategies, as well as the strengths that can be used in a combined solution to the proposed planning approach. As will be stated later on, the planning strategy proposed here is based on an HTN approach.

From the cognitive perspective, planning strongly depends on knowledge and understanding competences. This dependency is grounded in the tight coupling existing among knowledge and decision making. As stated by the authors of [39], there are some pragmatic concerns about this relationship that do not have a unanimous answer. What does an agent need to know in order to perform a specific action? When does an agent have to stop gathering information and make a decision? Or at what point does an agent have to answer “I don’t know what to do”? Answers to these questions lead to the conviction that some degree of common sense is required. Refer to [151] for a thorough analysis of the most relevant knowledge-based planning techniques available in the literature.

This chapter is structured as follows. Firstly, it is necessary to analyze the challenges that a planning strategy for Ambient Intelligence needs to face. The following section is concerned with the fact that planning also needs to be comprehensively organized, taking into account that different circumstances might arise, such as failing plans or multiple choices. These previous sections analyze the requirements that need to be faced by a planning strategy for Ambient Intelligence whereas the following ones address the proposition of the planning approach to cope with the stated requirements. Since it has been specifically designed for an Ambient Intelligence environments, general benchmark problems are not suitable for comparison purposes. Finally, interim conclusions are set out in the last section, where a comparison between planning strategies for Ambient Intelligence is also provided.

6.2 Challenges in planning for Ambient Intelligence

This section is devoted to describing how to cope with the challenges involved in planning under Ambient Intelligence. Some of these challenges are a direct consequence of the pervasiveness that characterizes those environments, while others are implicit in any planning strategy itself. The description of these challenges starts by addressing the requirements that need to be generally fulfilled, without considering at this stage any of the implications of common sense or the peculiarities of the targeted environments.

Additionally, on the basis of the identified requirements, the functional modules that are involved in the proposed planning strategy are described and additional implementation features are also provided. Finally, common-sense knowledge and environmental peculiarities are also considered as additional variables of the equation, and the requirements they pose are also analyzed.

6.2.1 Planning requirements

One of the first requirements that need to be satisfied before a planner can actually do its work is that of associating plans to goals. Each goal holds a list of plans that can be instantiated in order to achieve, perform, or maintain that goal. For example, if there is a goal intended to identify every person that goes into a specific room, there can be several plans worth mentioning for that purpose, such as the one that performs a facial recognition, or others based on iris, fingerprint, or voice identification. All these plans can be associated with the goal of identifying people. In any case, plans are also conceived under the spectrum of the goals that they serve.

The fact that plans are associated with goals means that whenever a goal needs to be satisfied, there is a set of plans that can lead the system to achieve it. In the most basic case, only one plan is provided for each goal, meaning that there is only one way of achieving the goal. Whenever more than one plan is associated with the same goal, all of them become candidates. If plans are not incompatible among themselves, several plans can be followed at the same time, otherwise, the detection of contradictions forces the planner to choose among the possible candidates the one with the highest chance of success. These sorts of decisions need to be stated beforehand, at a meta-planning level, as a means for plan conflict resolution.

Plans can therefore be graded according to how close they are to the goal pursued. There is, therefore, an implicit need for plan projection that allows the planner to predict the result of accomplishing a certain task, on the basis of knowledge about the state of the world and the tentative plans. Although explained in more detail later on, at this point it is worth pointing out that this activity implies some sort of planning under uncertainty, since incomplete knowledge forces the planner to make assumptions about unknown features. Resorting to default reasoning can lead the planner to the conclusion that by getting a snapshot from the camera located at the entrance of the room, the face recognition algorithm will be able to recognize the person. The planner is here assuming that the person will not voluntarily or involuntarily hide his face from the camera and that the light conditions are appropriate for capturing the person's face.

Nothing has yet been said about how goals occur to the planner, since it is not unique, but rather they can be activated by the planner itself, they can also be externally stated, or be derived from the detection of an unsatisfied need. The planner, in its endeavor to achieve a goal, can eventually observe the need for one or several sub-goals, that when satisfied lead the system to its ultimate goal. Moreover, goals can also be externally dictated, for example whenever a new system requirement comes into play. The requirement itself turns into a goal to be achieved. Finally, the maintenance goals category comprises those that rather than being dedicated to achieving a certain state, are intended to maintain a desired state. Whenever the system abandons that desired state, an unsatisfied need appears, therefore bringing about the goal to return the system to the desired state.

However, it cannot be forgotten that real system goals rather than arising in an isolated manner, exist concurrently. The interactions between those goals also need to be considered, since new constraints might come up on the scene. For example, consider a scenario in which one of the system's goals is to keep a room at a specific temperature due to the presence of temperature-sensitive devices. Additionally, one of the system's users requests the room to increase its temperature beyond the limits set by the maintenance goal. This scenario presents a situation in which several active goals are incompatible, and at least one of them needs to be dismissed for the other to be satisfied. Nevertheless, canceling interactions are not the only type of relationships that might arise among goals. The planner needs to characterize those interactions while also stating how to proceed depending on the detected type of interaction. In the foregoing scenario, the focus could be on trying to identify why the user wants the room to be at a higher temperature since risking the temperature-sensitive devices is not a choice. Roughly, two different types of goals can be identified, those that are necessary or mandatory and those that are desirable. The first should be decided before the second. This requirement poses the need to allow goal abandoning when unresolved conflicts are detected.

Sometimes, it is possible to consider these interactions in advance in order to adapt the available plans to avoid conflicts. For example, if the interaction between the aforementioned goals is considered in advance, different solutions might be adopted. In this sense, a plausible reaction can consist in minimizing the difference between the external and the internal temperature. By minimizing this difference it is therefore possible to minimize the user's desire for a temperature increase that compensates the cold felt. In this regard, it is therefore desirable that plans can be modified or generated on the basis of the existing goal interactions.

6.2.2 Planning from the human mind point of view

The previous subsection has presented the list of requirements that typically arise when building a planner. At that stage nothing has yet been said about the restrictions that might be imposed by the fact that such planning strategy should consider common-sense knowledge as a constituent part. Consequently, this subsection concerns those features of the human mind that help it succeed in performing planning activities. It is worth mentioning a paradigmatic example of a planning activity such as a chess game. It is a fact that humans and computers are decent adversaries for one another. However, the same cannot be said when the planning task takes place in the "*open world*" rather than in a closed dimension of it like in chess. In an open space dimension, different sources of changes coexist, and therefore, different desires and goals might arise other than those coming from foreseen sources.

It seems obvious that planning in an open world requires a higher level knowledge other than just extensive rules that determine an enclosed space like the chess space. Holding large amount of knowledge is nowadays a solved issue, however, the matter is not where to keep large amounts of information but rather, how it should be stored so as to easily enable its retrieval independently of how large the knowledge base becomes. In this sense, one of the main challenges to be faced consists in devising a means for efficiently retrieving information.

Observed from the point of view of the computational resources that are involved in retrieving and dealing with large amounts of knowledge, it is fascinating how successfully humans perform in doing so. Any attempt intended to imitate such activities requires serious computational resources. But there should be something else too, since it is obvious that computers are more computationally powerful than human minds, and in the light of the results humans seem to be resorting to some sort of short cut given their great success.

It is the extremely flexible structures in which human knowledge is held that, among other features, differentiates humans from computers. Humans are capable of dividing the large body of knowledge they hold into separate parts, and depending on the circumstances, they are also capable of efficiently splitting it into manageable small bodies.

An additional feature that also characterizes the human mind is related to the mental activities that so easily enable humans to infer new knowledge. These mental activities include those based on instinctive or learned reactions, deliberation, reflective thinking, self-reflection and self-consciousness, imagination, and projections, extensively studied by Minsky in [97]. But above all, it is the human capability to generate partial and provisional plans which makes humans so good at planning.

The fact that human plans are neither complete nor logically correct, rather than supposing a weakness in the human ability to plan is in fact what gave it with characteristic flexibility, which in the end is what differentiates human plans from computer plans.

In spite of its great strengths, human planning capability does also have some weaknesses. For example, there are some human mind limitations that prevent people from remembering specific facts that were known before. Even worse, the human mind can also incur in the error of considering previously known true facts as false instances.

Additionally, the everyday world can be seen as an *open world* in which different sources of change are concurrently taking place. This *openness* makes it difficult to characterize in advance the possible changes that might take place as a search space. On the contrary, the infinite list of possible states makes it unfeasible to address the problem of planning in an open world in the same way as planning in the chess world can be carried out.

This peculiarity also poses additional requirements that need to be dealt with in the same way humans do. *Conventions* [130] is one of those human mechanisms that simplifies the task of dealing with the real world. In this sense, conventions simply refer to those stated responses that are expected to take place given a certain set of preconditions. It does not mean that those conventions are mandatory on the contrary, it simply means that they are more likely to occur. If those conventions or expected practices are known beforehand it can help in reducing the effects of a changing world, since planning can be more easily performed in a reduced search space.

6.2.3 Functional units of a planning strategy

The previous subsection has described a comprehensive list of requirements that need to be taken into account when engaged in the task of building a planner. These requirements can be disassociated into different functional units as Wilensky suggests in [150]. Inspired by these directions, the following subsections described how the aforementioned requirements can be tackled from the perspective of the different functional units making up the proposed planner.

The Goal Detector

Plans are formulated in terms of the goals towards which they are directed, which means that plans come into play only when unsatisfied goals appear in the scene. In this sense, it is the Goal Detector's responsibility to determine whenever a plan needs to be instantiated in order to satisfy, maintain, or achieve a specific goal.

The Goal Detector characterizes those situations that are relevant to the planner and monitors them in seeking for arising goals. There are several situations prone to elicit new goals. One of those situations arises when changes affect the current state of the world. Changes that deflect the system from the desired state might motivate the system's need to return to the previous state, with that need becoming a new goal. Moreover, as a result of goal interaction new ones might be enacted.

Aside from those goals that have been *externally* provided to the planner, as general guidelines, there are some additional goals that concern the inner functioning of the planner. In this sense, there are some decisions that affect which plans are instantiated and how. The planner's internal structure deals with these decisions which, as will be explained below, fall into the *meta-planning* category aspects in charge of advising the planner about how to deal with goal and plan conflicts. Those directions are also goals themselves, since they address the planner's behavior on the basis of the goal sets that need to be satisfied at all times.

In the same line as the meta-planning directions, the planner also needs to hold some additional information, such as its set of preferences, dislikes, or needs. In other words, the planner's mental states also need to be considered for it to behave correctly. Along with the planner's mental state, it is also necessary to keep an updated track of the current state of the world. The combination of the state of the world with those planner mental states arise new goals whenever unsatisfied needs are detected.

The Plan Proposer

Goals can only be achieved by means of their associated plans, and therefore, goals need to be accompanied by the plans that might achieve them.

There are different possibilities for holding such plans. One possibility is to roughly describe the actions that comprise the plan, associating goal types to the plans of these plan-bases (in the sense of a knowledge base of plans). Those goals that respond to foreseen situations might be easily handled in this way. On the contrary, it might happen that as a result of a goal interaction or an unpredicted need a new goal is notified to the planner. In such situations, in which no plans have been associated with it, the Plan Proposer might resort to stereotyped solutions or to the use of analogies, proposing the same plans with which similar goals were addressed by adapting them to fit the existing differences with the similar goal. For example, the planner might have overseen the need for artificial light in a room during daylight hours; however, an eclipse of the sun create this unsatisfied need, which is working in an illuminated room. Resorting to a similar situation, at night time, light is provided by turning on the lights, and therefore, the same plan can be associated without requiring, in this case, any adaption to fit the active goal.

It might also be possible that none of the available plans lead the system to fulfill a given goal, independently of whether the goal has associated plans. Such situations require the Plan Proposer to account for a solution based on the analysis of how the available actions can be composed so as to lead the system to the desired or goal state. As discussed in previous chapters, these are the most common situations when dealing with everyday situations. As a summary of what will be described later on in this document, when available means[2] have not reported any suitable plan to proceed, an in-depth search can be performed upon the available actions, so as to seek those that when properly put together can lead to the desired goal. The proposed search is supported by a common-sense heuristic dismissing those actions that are not sensible or feasible from a common-sense point of view.

The Plan Projector

The Plan Proposer needs a means to evaluate the suitability of the available plans, so as to predict the convenience of undertaking a certain plan upon a different one. This functionality is provided by the Plan Projector.

This functional unit works as a simulation tool, in which given the description of the current state of the world, it performs the successive actions that comprise a given plan, under a close supervision of how the world state is evolving. For the sake of continuity, it is worth noticing that, internally, world state evolution can be tracked by means of the *possible world* theory, proposed in the previous part.

Although projecting plans might be seen as a trivial task when considered under static worlds, it is not so simple when extrapolated to real world. The *block world problem* is an example of a static world, in which there are no external influences affecting the state of the world. On the contrary, the everyday world is full of external dynamics that are continuously affecting the current state of

---

[2] Such as those based on stereotyped solutions or analogies.

the world. Some of those external dynamics are irrelevant for a certain plan projection, however, it is likely that most of them are not. Additionally, the consideration of all the possibilities that might arise at each stage of the plan projection is rather more than unfeasible, therefore a middle point solution need to be proposed. This middle point solution consists of focusing only on those undesirable elements, and what external influences in combination with the on-going situation might lead to them.

At some point, the projection of a given plan might create the need for new goals, and therefore, the need to identify how they can be satisfied. In such circumstances, the Plan Projector can recursively resort to the Goal Detector so as to determine how to proceed in the projection.

The Plan Executor

Once the plan has been devised, it needs to be undertaken, and this is the responsibility of the Plan Executor. Recall that nothing has yet been said about how to associate plans to means or actions since only the Plan Executor needs to know how to accomplish the actions that compose the plan. In this sense, it can be stated that there exists a functional decoupling which simplifies the way the modules composing a planner can be implemented.

This functional unit is also responsible for supervising the evolution of the plan's execution, in seeking for those undesired external influences that were foreseen by the Plan Projector, and reflecting the changes that the progress of the plan's execution causes to the world state. If new changes force the planner to revise its planning strategy it is the Plan Executor's responsibility to notice such deviation and react in response to it.

6.3 Planning the planning

The last issue that needs to be considered before accomplishing the task of proposing a planning algorithm for behavioral response generation is to determine the plans and goals that are to drive the planning mechanism itself, also known as *meta-planning*.

As mentioned above, goals rarely occur in isolation, and for that reason the first thing to determine is the nature of the relationships that affect goals. Additionally, the circumstances surrounding goal interactions might also be relevant for the sake of resolving possible conflicts or clashes. The taxonomy for goal relationships proposed by Wilensky [150] is used here as the reference guide upon which to support the planning strategy followed by the planner described in this thesis.

Wilensky therefore identifies four categories of goal relationships, namely: goal conflict, overlap, competition, and concord. At the same time, those categories can be grouped as negative interactions including both goal conflict and competition, and positive interactions such as goal overlap and concord.

Negative goal relationships can be caused by situations in which a common limited resource is required by several plans that are being implemented for different goals. Think of the mutual exclusion problem when it comes to shared hardware.

Goals that lead to states impossible to occur simultaneously also lead to conflicting goals. The room temperature example presented above poses a situation in which two different states cannot take place simultaneously, like the room being at two different temperatures[3]. Also, primary goals need to be preserved above those that have been considered to be additional or secondary. Continuing with the same room temperature example, preserving device integrity is considered a primary goal in regard to the secondary goal of providing the desired temperature.

---

[3] Except for any question regarding different scale values

The means proposed for dealing with goal conflicts basically fall into two groups, continuing with the directions provided by Wilensky, namely: re-planning and changing circumstances. In the first case, the final goal is to find a different plan that leads to the same goal although avoiding the conflict, while the second strategy consists in trying to change the circumstances that led to the goal conflict. The later strategy requires the state of the world to evolve to a different situation in which the conflicting issue disappears. If none of those mechanisms resolves the goal conflict, then it is time to evaluate goal abandonment. Provided that goals have been classified by their importance, primary ones have a higher preference than secondary ones. Whenever goals of the same precedence are being evaluated the decision of which one to be dropped is founded in the plan projection, considering needs, likes, and dislikes. Plan projection can also disclose those cases in which a partial satisfaction of one of the goals is possible, in contrast to its full abandonment. Finally, if planner supervision is considered, those goal conflicts that cannot ultimately be resolved by any of the previous means, can resort to an external resource to decide how to proceed.

6.4 The planning strategy

So far, this chapter dealt with those aspects that affect planning, although avoiding the details that concern how to undertake the construction of a planning strategy for Ambient Intelligence. This section is dedicated to that particular concern.

The peculiarities of the Ambient Intelligence scenarios, in which the physical world that surrounds it is prone to cause unforeseen interactions, suggest the need for a planner that implements a "*divide and conquer*" strategy. Plans cannot be considered as static instances since their execution might lead to situations rather different from the one the planner expected to achieve, mainly due to the changes that take place in the context and that cannot be foreseen. The strategy proposed in this thesis consists in addressing the planning task from two different perspectives. The proposed approach therefore consists in devising the course of action that might lead to the targeted state and then, identifying the means that can be used to implement those actions. Both approaches are here referred as *skeletal planning* and *action planning*. The skeletal planning is the planning strategy analyzed in this chapter, since it falls into what so far has been referred to as *behavioral generation*. The action planning is described in the next chapter as part of the *behavioral implementation*.

6.4.1 General features of the planning approach

The skeletal planning approach, as for any planning strategy, starts by identifying the planner goal and then associates plans to goals. The main difference is that in this approach, rather than considering plans as complete recipes, they are basic skeletons that specify the abstract structure that the final plan will have to satisfy. For example, one of the goals of a surveillance environment is that of identifying an intruder whenever an unauthorized presence is detected. This approach consists in providing plans as abstract formulae rather than specifying the static course of actions that lead the system to the identification of the intruder. Recall the many different possibilities that might arise when trying to perform an intruder identification, independently of the approach followed.

The physical world interactions require a planning strategy that acts in such a world to devise a *one-step-at-a-time* plan. The skeleton plan simply states that a feasible means of identifying an intruder is to perform a biometric identification upon any biometric feature that becomes available. The evaluation of the world state will provide feedback about how to properly perform such a task. For example, it would be possible to take a snapshot of the intruder, detect his/her face, and try to get a match using a facial recognition service. Obviously, for this plan to be successful many considerations need to be taken into account, such as the fact that the person needs to be in front of the camera or that lighting conditions be appropriate for capturing a good picture. Aside from such conditions, nothing has yet been said about the availability of the service, an aspect that has been left for the following chapter. In any case, this example demonstrates the unsuitability of traditional planning strategies when the real world is involved. It is necessary to adept the planning strategy to the peculiarities of planning under uncertainty, non-determinism and stochastic space.

Physical world interactions cannot be foreseen and therefore prevent the Plan Projector from knowing the specific result of accomplishing a specific plan. Evaluating the plan performance requires the planner to evaluate the world state at each plan stage, revising the plan when physical world interactions deflect the plan from its expected state.

Ideally, planning is based on knowledge about the current state of the world, but when the physical world is involved this premise needs to be revised. In the block worlds, there are no external influences, and therefore the state of the world is always known. On the contrary, the open world makes it unfeasible to control all the different possibilities. LaValle in [75] points out that the way to overcome this shortcoming is by changing the state space to consider *information* rather than the state of the world. The *information space* is composed of the history of sensor observations, actions and initial conditions.

6.4.2 Defining the planning problem

Every planning strategy has a set of common elements that define the characteristic of the planning problem. Recall that these elements correspond to the planning problem at the upper layer, basically intended to devise those actions that could lead the system to its ultimate goal, yet avoiding the considerations about how to implement the given plan.

The first of the elements that define a planning problem is the *state space* $\mathcal{S}$. This element describes all the states that the planner can be in. For example, in the block world problem, the state space is determined by the description of all the possible combinations of the block positions. Nevertheless, as has been already stated, whenever the physical world is considered it is hard to determine the different states that might be involved. For that reason, this problem resorts to an information space, over the use of a state space. The information space considers the information gathered from the environmental sensors as well as the actions and additional observations that can be retrieved from the environment. Such information can be considered as part of on-going situations, in the same sense as the situation calculus. The infinite number of situations that might arise in such contexts makes it unfeasible to consider a situation space. For the planning problem considered here, states are depicted as a tuple of the form $s = (a, o)$ meaning that the situation $s \in \mathcal{S}$ is the resultant state after performing the action verb $a \in \mathcal{A}$ upon the given $o \in O$, where O represents the set of objects in the world.

The second element of a planning problem consists of the set of actions or *action space* $\mathcal{A}(s)$ available at each given state $s \in \mathcal{S}$. The reader might believe that the action set is composed of those actions that are directly provided by the available services. Rather than constraining the action set to those that are directly provided by the services and devices of the context, all the actions considered at the common-sense knowledge-base have been taken into account. The fact that the planning problem has been divided into two layers allows the skeletal planning to be concerned only with the actions that lead to the goal, overlooking other aspects regarding how they can be implemented. These aspects are therefore relegated to the action planning layer. This twofold approach is the cornerstone of automatic service composition, so action planning is intended to overcome the service shortage by resorting to available services for a different end than the one they were conceived for. For example, recall the foregoing example that used the fridge light to illuminate a dark kitchen while the kitchen light bulb was being replaced.

An additional element is that of the state transition function f that, given the current state or situation and the action space, produces a new situation for every action, out of the space action, that is available at the current situation.

$f(s,a,\theta)$ for $s \in \mathcal{S}$, $a \in \mathcal{A}$ and $\theta \in \Theta(s,a)$

Function f returns the actions, from the action space, that are available in the current situation s. For the planning problem considered here, availability is determined by the satisfaction of the pre and post-conditions of each of the considered actions. Action unavailability means that a certain action either leads the system to a situation incompatible with the desired goal state or, on the other hand, it requires a different state of the world from the current one in order to take place. The Θ function therefore provides the set of situations that can be reached given the current state and the execution of any of the actions that are available at the same state. Recall that the transition function notion is equivalent to that of *situational fluent* from the situation calculus.

Stages, denoted by $k \in \mathcal{K}$, also need to be considered so as to conceive the plan execution as an incremental task. Moreover, stages are used by the planner to evaluate the evolution of the plan execution, identifying possible plan deviations.

In the state space, there is one that is especially relevant, which is the *goal state*, denoted by $\mathcal{S}_G \subset \mathcal{S}$. The goal state is no unique, there are as many as the number of goals ascribed to the system.

Finally, it is necessary to have a function that evaluates the goodness of an action in comparison with the others. The cost function L weights, given the current and the goal situations, the suitability of each action, in seeking the course of actions that maximizes[4] the cost of reaching to the goal state. The history of states, actions, and available actions are correspondingly denoted by $\tilde{s}_K, \tilde{a}_K, \tilde{\theta}_K$, so the cost of a given course of actions can be calculated as follows, given that F is the goal state:

$$L(\tilde{s}_K, \tilde{a}_K, \tilde{\theta}_K) = \sum_{k=1}^{K} l(s_k, a_k, \theta_k) + l_F(s_F) \tag{6.1}$$

The cost function we propose here seeks to weight the different alternatives by measuring the action that is closest to the targeted action. However, how can distances between two action verbs be assessed? A possible solution could be based on the use of *n-gram* models. In fact, bigrams could be used to evaluate the distance between an action and the target.

A bigram model is a probabilistic model used to predict the probability of a word appearing, given that a certain word has appeared before. This probabilistic model has been extracted from analyzing large bodies of text, from which grounded probabilities can be taken. Recall the fact that the semantic model that is behind this work considers actions and objects upon which those actions are performed as indivisible tuples. These tuples can be evaluated as bigrams measuring the probability of a given action being performed upon a given object. Moreover, actions among themselves also need to be ranked, using for this purpose an action n-gram model. Two n-gram models are required to evaluate the goodness of the available choices, one only for actions, and an additional one for actions and objects. The cost function for the Ambient Intelligence planning problem is the following:

$$l(a_k, o_k, \theta_k) = P(\theta_k | a_{k-1}).P(o|\theta_k) \tag{6.2}$$

6.4.3 The planning algorithm

After having enumerated all the elements that are involved in a planning problem, this subsection now deals with the planning strategy to be followed so as to achieve to the goal state. In order to do so, we

---

[4] In this case, the optimum value is that of the greater value, since as explained later, the proposed approach deals with probabilities

are proposing a Dijkstra-like algorithm in which each stage of the execution of the plan is expected to be closer to the goal than the previous stage.

Recall that the state space is neither known nor discrete due to the interactions with the wider world that surrounds Ambient Intelligence environments. In contrast to the traditional Dijkstra algorithm, states cannot be weighted beforehand, but will have to be evaluated at each stage of execution. Moreover, revisiting nodes is not avoided because states are described in terms of the situation that results from performing an action upon a given object. Sometimes it might be desirable to perform the same action several times upon the same object.

Algorithm 1 Skeletal planning(s_0, s_g)

```
$\pi = (\mathcal{A}, O)$
$s_0 = (a_0, o_0)$ that have arisen the goal $s_g = (a_g, o_g)$
$s_c = s_0$ $a_c = a0$ and $o_c = o_0$ current values are the initial values
while $s_c$ is different from $s_g$ do
    $\theta = \Theta(a_c, o_c)$
    while $\theta_i$ do
        $o_\theta$ = perfroms_upon($\theta$)
        while $O_j$ do
            $l_i = P(\theta_i|a_c).P(o_j|\theta_i)$
            if MAX($l_i$) then
                assert to $\pi$ the values $(\theta_i, o_j)$
            end if
        end while
    end while
end while
Recursively call the Skeletal planning with the values in $\pi$
Add $\pi$ to $\Pi$
Return $\Pi$
```

Basically, the planning algorithm proposed in Algorithm 1 consists in a recursive version of a Dijkstra-like algorithm. The algorithm is instantiated with a pair of states or situations, in which the first situation matches the current situation and the second one matches the goal state. Note that a situation is also represented by means of a pair of values corresponding to the action and the object upon which the action was executed. For instance, a situation that gives rise to an unsatisfied need is that of detecting unauthorized presence in a certain location. This situation is described as $(detect, unauthorizedperson)$ which simultaneously leads the system to try to achieve the goal of performing an intruder identification.

The current state or the s_0 state is the situation that has arisen after an intruder had been detected. Using the planning algorithm notation, the situation corresponds to the tuple $(detect, unauthorizedperson)$. Additionally, this type of situation creates a goal to be achieved, which is the s_g state, denoted by the pair $(identify, intruder)$.

The first stage consists in determining those actions that are available in the current state, by means of the Θ function. As mentioned above, the first thing the Θ function does is to list the properties of the world that result from achieving the goal state. After that, the action space is explored seeking those actions whose effects are equivalent to that of the action goal. All the actions that are capable of producing the same effects are returned in the θ set. Afterwards, each of the available actions is analyzed from the point of view of the objects they might act upon. For each object the probability of both worlds is also computed. Finally, the action and object pair that reaches the highest weight is the next step selected in the plan. The same algorithm is recursively instantiated with the selected state, and the same process is carried out until one of the states matches the goal state.

6.5 Interim conclusions

This chapter is in the first of two, intentionally addressed to describing the method for enabling an Ambient Intelligence system to display goal-driven autonomous behavior. In order to achieve this, systems need to be enhanced with the required capabilities so as to generate behavioral responses according to the goals pursued and the current situation of the context.

From the perspective of automatic service composition, this chapter has been devoted to determining the set of actions that will be part of the composite action. This strategy has been described as a planning task, in which actions and objects in the real world are the means used to devise the most suitable way of achieving a desired state given the current situation. The following chapter is concerned with how to implement these actions by means of the available tools in the form of services and devices deployed in the supervised context.

The main contribution of this chapter to the overall thesis is the planning algorithm proposal. This algorithm is referred as the *skeletal* planning algorithm due to the fact that it is only intended to outline the structure of the future action plan. As part of the first stage of the service composition, this algorithm is only responsible for enumerating the actions that are part of the course of actions, also referred as the composite service.

The action planning layer is based on these skeleton plans in order to implement a heuristic search upon the service space, seeking those that can finally fulfill the desired goal state. Search and inference activities are carried out using the common-sense knowledge held at the knowledge-base system described in previous chapters. .

Chapter 7

Behavioral Response Implementation

Action has meaning only in relationship and without understanding relationship, action on any level will only breed conflict. The understanding of relationship is infinitely more important than the search for any plan of action.

– *Jiddu Krishnamurti*

Summary – *This chapter complements the previous one, in which the behavioral response process was partially described, by detailing how to map those skeletal plans onto specific instantiations of services and methods. The proposed approach is also concerned with the device dynamism aspects that characterize an Ambient Intelligence environment. The flexibility required to overcome this issue is achieved by means of an automatic service composition strategy. Understanding service capabilities is essential for the service composition task, since it is grounded in the possibility of using a service for a different purpose from the one that it was originally conceived for.*

7.1 Introduction

The preceding chapter proposed a methodology for generating generic behavioral responses, on the basis of a planning strategy, and motivated by the detection of unsatisfied needs and the awareness of the goals to be pursued or maintained. Basically, the proposed methodology consists of a goal-driven planner, whose main purpose is to put into action the skeleton plan that leads to the goal state.

There exists a relationship between these two chapters, in the sense that the result yielded by the behavioral response generation (BRG, from now on) module needs to be refined by the behavioral response implementation (BRI, from now on) module. The focus of this chapter is aimed at describing the foundations of the BRI module.

The input to the BRI module consists of a pair "*action - object*", which mean that the skeleton behavioral response consists in undertaking the given *action* upon the specified *object*. It is the BRI module's responsibility to devise the course of actions that, implemented on the available services, are capable of achieving such a generic intention. For example, the previous chapter explained how, after an unauthorized presence was detected, the system's goal was to try to identify the intruder. At a first stage, the BRG module yields the skeleton plan that allows the system to vaguely *identify* the *intruder's biometric feature*. The BRI module goes a step further and is engaged in turning the vague plan into a set of more specific course of actions.

In this sense, starting from the skeleton plan, the BRI module is responsible for reaching a solution that, for example, foresees that it is possible to use an intruder's picture to perform a facial recognition of it. It is also possible to try to identify any of the electronic devices that the intruder might be carrying, and then try to match the device ID to any of the known ones.

The main strength of an approach that dynamically constructs plans out of vague directions consists in its low-coupling with the services that are going to implement the plan. Decoupling plans from the services that will be implementing them enables a successful addressing of the device dynamism feature that characterizes Ambient Intelligence environments. Vanishing services do not affect the system's capability to generate behavioral responses since the same skeleton response can be made specific by means of the remaining services available at that moment.

Aside from the decoupled responses, an additional benefit from the proposed approach is that of dynamically computing the set of services whose actions best fit the requirements stated by the skeleton plan. If plans were statically attached to a fixed set of services, it will not be possible to make the most of the appearance of more suitable services, or to overcome the disappearance of services. On the contrary, implementing a strategy based on a service composition approach works as a service *overload* that intentionally broadens the range of functionalities provided.

It has to be highlighted that here a distinction is made between service composition and service combination, in the same sense as it is a fact that *combining music* differs from *composing music*. The latter requires understanding additional aspects such as harmony, rhythm, chords, whereas music combination seems to refer to the union of a sequence of musical pieces. The foundation supporting the distinction is that of the different skills that are involved in both processes. In the review of the literature it is common to find approaches that claim to perform service composition whereas they are simply sequentially combining services. Note how composition involves combination, but is not restricted to it. To date, among all those that claim to perform service composition few of them are really addressing the challenges that are involved in composition.

The approach presented here makes the distinction between composition and combination and is devoted to addressing them both. Service combination is used for simple plans, when available services fulfill the skeleton plan requirements. On the contrary, composition is demanded when available plans do not seem to fulfill, at first sight, the plan requirements, having to overload the service functionalities. Service overload is grounded in the common-sense knowledge that describes how actions affect the world. It is therefore possible to identify those services or actions that, despite not being equal, are capable of producing the same effects in the context.

After having stated the interdependence relationships established between this chapter and the previous one, as well as the difference between service composition and combination, the next step consists in revising the different approaches found in the literature intended to support service composition. The rest of the chapter is intended to describe with a higher level of detail the challenges that arise when trying to support service composition. The next section concerns how to implement the service composition approach by means of a planning strategy. Finally, the last section outlines the main conclusions drawn from the proposed strategy for behavioral response implementation.

7.2 Approaches to service composition

Many proposals have intended to provide a solution to ease the process of developing applications for Ambient Intelligence. Some of the most relevant contributions are listed bellow, with the intention of providing a global vision of the evolution towards service composition.

- **Configuration files:** The first attempt to provide some sort of dynamism was based on the adaption concept. This proposal was intended to provide architectures with adaption capabilities by means of configuration files. These architectures counted on some features that could be customized in order to select the components loaded at startup time. This approach allows a limited level of dynamism, far from what an adaptive middleware should be, since the main features are placed at the middleware kernel and cannot be changed. Therefore, rather than adaptive middlewares they should be considered customizable middlewares, supporting just a fixed group of cases.

- **Reflection:** As a second attempt, after the use of configuration files, reflectiveness appeared as the solution to add some dynamism support to middlewares. This proposal advocates a middleware core with a minimal set of services installed in devices. By means of reflective mechanisms, applications can obtain context information from the middleware, and use it to tune the middleware behavior.

- **Reflection and metadata:** The next stage in this development is based on the combination of reflection and metadata, aimed at developing adaptive and context-aware applications. This approach is mainly based on policies, that is, the use of a set of primitives aimed at describing how the context might change and how these changes are to be treated. Since conflicts between policies may arise, a solution based on a micro-economic approach was proposed in order to handle this conflicts.

- **Externalization:** Although reflective middleware services do support configurability, by supporting replacement and assembly of components in reaction to changes, the reality was that most of them assumed a basic backbone of fixed services. The externalization approach suggests the need for a middleware architecture that explicitly externalises the state, the logic, and the internal structure of middleware services, in such a way that the system can be updated, upgraded, or change its configuration without requiring user intervention.

- **Policies:** This approach proposes the use of profiles, where the associations between services and policies applied to these services are described. Profiles are passed down to the middleware, and whenever a service is invoked, the middleware consults the profiles of the application that requests it. The profile determines which policy can be applied in the current context, depending on the state of the requested resource, thus relieving the application from performing these steps.

- **Web Services:** Among all these different approaches towards service composition, Web Services have been by far the most popular. This XML-based approach allows the specification of web services that can be dynamically loaded according to the requests. A service is specified by means of a service's abstract interface and the non-functional properties associated with the service. This approach provides a set of Service Repositories containing information about local and remote service repositories.

- **Planning:** Artificial Intelligence planning techniques have also been used for service composition since the task of composing a service can be viewed as a planning problem in which the goal state is the composite service, and the actions that cause the state transition are made from the available services. Algorithms such as those based on an HTN-like approach recursively decompose the targeted composite service into sub-tasks, until reaching those that can be directly instantiated by basic services.

- **Software Agents:** The suitability of software agents for decomposing the complexity of a problem into smaller entities can also be exploited for service composition purposes. In a Multi-

Agent System (MAS), individual agents can be in charge of specific functionalities, which can be used as constituent parts of composite functionalities.

- **Workflows:** This approach shares many features with the Web Service approach. The role of the service provider is to be in charge of describing the *workflow* (service inputs, outputs, and external dependencies) of service instantiation comprising a composite service.

- **Ontologies:** Finally, it has to be pointed out that nowadays, the use of ontologies is gaining attention. Among all the ongoing proposals on these field, *domain ontologies* is one of the most relevant, intended to model the domain knowledge and provide semantics to service description. The capability of expressing semantic relations among services is quite useful in guiding the composition process.

As has been already stated above, this thesis advocates the planning approach for implementing the behavioral response outlined by the BRG module. Further details about the proposed planner are provided bellow.

7.3 Service composition challenges

Before undertaking the description of the planning strategy, it is necessary to analyze the challenges that have to be faced when trying to design an approach for service composition.

Traditionally, the service composition task needs to be provided with both the service description and the binding information. The service description simply provides the information that supports the service identification whereas the binding information states those requirements that need to be satisfied when the service is part of a composition, for example protocol-dependent details. Think of the web services examples, where the WSDL (Web Services Description Language) is used for describing the service. The service description involves the statement of the messages that can be exchanged with external services, as well as the binding information that contains the protocol specific details.

In addition to the service description, there are some other general requirements that need to be fulfilled by those services that are to be considered for composition tasks. Without yet focusing on any specific composition approach there are a set of general requirements that all the approaches need to comply with and that are listed below:

- **Basic services:** Composition process uses basic services as its raw elements. Therefore, the list of those basic services that are available needs to be known for them to be considered for the composition task.

- **Service Discovery Protocol:** Due to the fact that device dynamism is an issue that needs to be present in any attempt to compose services in Ambient Intelligence, there should be a mechanism that enables upcoming services to be discovered.

- **Automatic binding:** Device dynamism is not the only issue that needs to be considered under Ambient Intelligence, but also service heterogeneity. Different technologies, protocols, or different service features, make the service binding unfeasible without automatically abstracting those issues from the composition task.

- **Uniform treatment of services:** As a direct consequence of the above requirement, despite their heterogeneous source, all services have to present the same structure, as if they were being provided by the same service source.

- **External automatic service adaptation:** At some point it can be necessary to carry out an adaptation process during the composition, since external source services may differ in the interaction models and protocols. It is necessary to mask out these differences by means of translation and interface masks.

- **Dynamic service ranking:** More than one service can be suitable for a request, although just one is selected. It is necessary to effectively evaluate the suitability of one over the others. This evaluation is dynamic, which means that available services needs to be evaluated each time they are considered as part of a composite service.

- **Service continuity:** Also, as a result of the device dynamism new services might arise that are more suitable than the one being used in an already composed service. In such a situation, the previous service might be discarded in favor of a new service, which has achieved a higher mark in the ranking.

Aside from the requirements listed above, which refer to general aspects of the composition task, there are some additional issues that need to be considered for the particular task of automatically composing services.

Automatic service composition needs to be prepared to identify the need for a composite service, without explicitly receiving any *composition request*. The only sign that evidences the need for a composition process is the request posed by the BRG module. As has been already stated, the BRG module provides the BRI with a pair of the form "*action-goal*" that consists of the composition request that the BRI is engaged in. In other words, the BRG module detects the need for a behavioral response, based on the knowledge about the goals that the system needs to reach or maintain.

However, the only heuristic that the BRI module holds about the composition request is that of the high-level action that needs to be performed upon the given object. However, there is nothing like a work-flow or similar stating the services that are part of the composite one, or how to connect service inputs to outputs. Since it is an automatic approach, it is the role of the composer to determine which services better address the targeted high-level action.

Additionally, it has been claimed at the beginning of this chapter that the BRI module is supported in the common-sense knowledge as a means of heuristically addressing the composition process. It is therefore necessary to base the composition process on the common-sense descriptions of actions and events. However, further issues needs to be considered to provide an approach that complies with the common-sense key issues. Refer to Chapter 3 for a thorough description of those key issues of common-sense.

The semantic model proposed in Chapter 5 describes actions in regard to the changes they produce in the context. Conceiving actions in this way leads the proposed strategy for automatic composition to seek those actions that produce changes whose combination result in the same effects as though the *action* provided to the BRI had taken place.

Overall, these are the challenges that need to be faced in order to address the automatic service composition process from a common-sense perspective. Given these requirements, the following section deals with the implementation of the proposed solution.

7.4 Action planning for automatic service composition

The previous section has described the main challenges that need to be addressed by the BRI module, which is in charge of implementing the behavioral response outlined by the BRG module. Since the requirements stated for the BRG module match up with the theoretical capabilities of automatic service composition, the BRI module has been proposed as a service composer.

Among the many different approaches that can be adopted for service composition, in this thesis, a planning approach has been selected. Moreover, the divide and conquer strategy followed here, consists in dividing the behavioral response functionality into two modules, one intended to outline the most appropriate behavior and the other one intended to implement the directions provided by the first. Recall that this chapter is devoted to addressing the requirements and challenges involved in the second module, the one intended to implement the behavior outlined by the first module.

The main consequence of this divide-and-conquer approach is that at each layer different issues can be more specifically addressed. Recall how the BRG module is concerned with outlining the general features expected to be caused by the behavioral response. However, at the BRI layer, the focus is on devising the set of services that, when appropriately combined, performs the set of actions that produce the changes that drive the environment to the goal state. Nevertheless, in spite of the traditional planning approaches, it is not necessary to identify the goal state because it is provided by the BRG module. Recall that the *composition request* is stated as a pair of the form *action-object*. It is the responsibility of the BRI module to determine the most appropriate set of actions whose effects are equivalent to those described by the targeted pair *action-object*.

The composite service that matches up the effects of the desired goal state needs to be provided in such a way that the actions involved can be automatically and autonomously instantiated, without requiring user intervention. Once again, on the basis of the semantic model that underlies the proposed approach, the composite service is provided as a sequence of tuples of the form *service-action-object-time-location* with the associated meaning of the given *service* performing certain *action* upon the specified *object* at a specific *time* and at a given *location*.

This plan is then provided to the *plan executor* which takes care of accomplishing the instantiation of the provided services, which in turn, performs the desired actions upon the cited objects. Time and location are part of the plan, however, they are not always relevant for the plan instantiation and they can be omitted whenever they do not make any difference.

This comprehensive view is explained with further details in the following subsections, which describe the responsibilities of the composing units of the proposed planning approach.

7.4.1 The Goal Detector

The Goal Detector unit is responsible for detecting unsatisfied needs or goals. This unit is therefore expected to supervise the environment in seeking for those situations that have deflected the environment from its goal state, or when a new need has arisen. The Goal Detector is aware of all the goals that need to be fulfilled, satisfied, or maintained, and it is also capable of detecting any deviation from these stated goals. Nevertheless, the responsibility of the Goal Detector for this BRI layer is quite limited since the main responsibility of detecting the deviation rests upon the BRG layer, presented in the previous chapter. The Goal Detector of the BRI module is driven by the results obtained from the upper layer, and therefore, the BRG results are the only goals that need to be satisfied.

The Goal Detector unit does not need to supervise the environment in seeking for goal deviations since this information is directly provided by the BRG module. The *action-object* pair outlined by the BRG module consists of the goal state that needs to be achieved by the planning strategy implemented by the BRI.

The Goal Detector unit can be implemented in a reactive way since, as stated above, there is no need to supervise the environment because the BRG module provides it with the goal to be achieved. Therefore, the BRI module can be idle while the BRG has not detected any deviation from a goal or unsatisfied need that requires from it a behavioral response. Whenever the BRG module has detected the need for a behavioral response, it directly instantiates the Goal Detector unit of the BRI, providing it with the pair *action-object* that the BRI needs to fulfill by means of a composite service. The Goal Detector unit of the BRI module is in charge of instantiating the functionality of the following modules, which also wakes up in a reactive way when the Goal Detector receives the notification of the composition request. Note how *composition request* and *goal detection* are used indistinctly for the service composer implemented by means of a planning approach.

Recalling the example used in the previous section, the BRG module outlined the need to perform the action *identify* upon the *person identity*, as a solution toa new situation in which a restricted space has been accessed without the required permissions. The pair *identify - person identity* therefore represents the goal state that the BRI module is engaged in satisfying, in such a way that the resulting course of actions that compose the plan should provide the same effects as the *identify* action when applied to the *person identity* object. The Goal Detector therefore invokes the Plan Proposer and provides it with the targeted goal that consists of the pair *identify - person identity*.

7.4.2 The Plan Proposer

The fact that actions have been described under a common-sense perspective implies that the effects of some actions cannot always be devised beforehand. Some of these effects are context-sensitive, and therefore they depend on the current state of the context, which cannot always be determined beforehand. Additionally, some other effects are also dependent on the actions that are concurrently taking place, and therefore, their effects cannot be devised beforehand either, because the actions that could take part in the context, at the same time, are not predictable.

It has already been justified that the most compelling approach to build human-like behavioral responses is by describing actions from the common-sense point of view. This requirement therefore imposes the restriction that plans cannot be worked out just attending to the direct effects of actions since those effects are meaningless and unrealistic if the rest of the elements involved in the context are not comprehensively considered.

In contrast to traditional approaches to planning, in which the state space is discrete and known beforehand, the service composition problem poses a continuous space state, whose extension makes it unfeasible to be globally considered. The planning strategy cannot be therefore intended to propose, in isolation, the course of actions that lead to the goal state, because the dynamics of the surrounding world should be taken into consideration. The real world is affected by external and unexpected interactions that affect it and make it unpredictable. A planning strategy can therefore only be realistic if it is continuously re-evaluating the suitability of the action devised to lead to the goal state. The role of the Plan Proposer consists not only in the plan proposition, but also in adapting the plan at each stage of the execution.

The first step of the Plan Proposer's responsibilities is to provide a first draft of the plan that can lead to the goal state. At this stage, the first draft ignores the environmental dynamics, although it does consider the different types of action effects (delayed, indirect, concurrent, etc.). The draft produced is then provided to the plan executor, which is in charge of executing the actions involved in the plan and also of providing the feedback to the Plan Proposer which, is finally responsible for evaluating the expected results in comparison with the ones obtained. Different results force the Plan Proposer to recalculate the plan from the exact point from which the inconsistency was detected. Additionally, as will be explained shortly, it is the responsibility of the Plan Projector to determine the effects of actions, considering the common-sense description of actions and the additional information held at that moment about the context and the concurrent actions.

The action planning algorithm

Making the most of versatility of the services, systems for Ambient Intelligence could respond to whatever needs might arise, based on the available services and devices. In this context, needs are considered to be the desire to perform actions upon objects. To this end, the idea of Hierarchical Task Networks (HTN) is adapted to work with actions, instead of tasks.

The actions that can be performed by a system are determined by the devices and services available at each moment in time. Those actions that cannot be performed, due to the lack of services that provide a specific functionality, are named here as *non-feasible actions*. Whenever the system demands the execution of a non-feasible action, the planner comes into play.

As listed bellow, the `Planning` algorithm starts with an empty plan, the Π plan, to be completed with the list of actions, which at the same time are provided by services. This course of actions is intended to emulate the required non-feasible action. The course of actions is provided as a set of actions performed upon objects *A* and *O* respectively, and the results *R* of accomplishing such actions.

Algorithm 2 Planning(Π, A, O, R)

```
 1: $\pi = (A, O, R)$
 2: if $A$ is non-feasible then
 3:     get all the actions $A = (a_1, a_2, ..., a_n)$ that have the same result $A$
 4:     while $a_i$ is non-feasible do
 5:         delete $a_i$ from $A$
 6:     end while
 7:     while feasible action $a_i$ does not have an equivalent target object do
 8:         list all the objects $Objects = (o_1, o_2, ..., o_n)$ of action $a_i$
 9:         check if those $o_i$ are equivalent to or can be $O$
10:     end while
11:     Recursively call $\pi = Planning(a_i, o_i, resultOf\ a_i)$
12: end if
13: Add $\pi$ to $\Pi$
14: Return $\Pi$
```

In line 3 of the planning algorithm those actions whose effects are equivalent to the targeted action are selected. It is the Plan Projector's responsibility to evaluate the effects of each action under the current context situation. Among the selected actions, lines 4-6 get rid of those that cannot be directly instantiated by any of the available services. The algorithm therefore uses the term *non-feasible actions* to refer to those that, under the spectrum of the available services in the current context, are not directly provided by any of these services. Upon each of the *feasible actions*, lines 7-10 are devoted to identifying the object upon which the given action can be performed which, at the same time, is equivalent to the targeted object. For example, a *fingerprint* object is equivalent to a *biometric feature* object. This sort of equivalence needs to be identified. Finally, line 11 recursively instantiates the planning algorithm in trying to fulfill the action requirements, by accommodating the context conditions to those in which the action can take place. Finally, only when all the actions that are part of the plan can be instantiated will the planning algorithm abandon the execution, giving as a result, the course of actions which makes up the plan.

In its first instantiation, the plan that results from the execution simply consists of a draft plan that, is likely to be appropriate only for the first action, requiring the plan to be recalculated after the execution of the first action. For example, consider the situation that requires the *identification* of a *person identity*. The plan proposer yields a plan that roughly consists in taking a snapshot from a camera service, from which a person's face can be detected and a facial recognition service can be undertaken so as to identify the person. However, this plan, as it has just been described, is likely to fail. Think, for example, that in order to succeed the camera snapshot needs to capture the person's face, which is useless if the person is not facing the camera. The plan evaluation should reveal this inconsistency between the expected result from the *capturing snapshot* action, that is, capturing the person's face, and the result obtained, in which no face has been detected. The plan reevaluation focuses at this point on capturing the person's face. For example, if several cameras capture different angles, snapshots can be taken from each of them trying to detect the person's face from any of them. If it is not the case and there is only one camera available, the Plan Proposer should determine an additional plan to overcome the failing result obtained from the existing snapshots. The Plan Proposer now evaluates an alternative plan to capture the person's face. It is a requirement for the *capturing* action that the object to be captured is within reach of the capturing entity, in this case, the camera, whose reach is represented by the capturing angle. Recursively, the goal state is now turned into that of *moving the person's face within range.*[1].

An action planning strategy in which common-sense knowledge about actions has been stated reveals its great potential when subtle features play a key role in solving a problem such as the one intended to move within a given range of a person's face. A common-sense description of the *make noise* action reveals that an indirect effect of the action is that animals tend to face the source of the noise as a prevention of a possible attack. In the same way, humans try to identify unexpected noises by facing the source of the noise as an uncontrolled reaction grounded in our instincts that protect us from dangerous situations. The sole consideration of the direct effects of actions will have overlooked this subtlety of the *making noise* action, which seems to be the most suitable solution to achieve the targeted goal.

The Plan Proposer has therefore reevaluated the plan, and has added an additional action, which is the *making noise* action performed upon the *capturing angle*, so that as a result of such action the person involuntarily faces the source of noise, and a picture can be taken of his/her face.

The new plan is provided to the Plan Executor which is in charge of accomplishing it. For the sake of clarity, how the planning algorithm is also responsible for working out how to make noise from a specific place has not been described in detail. For this demand, whatever action can be performed in that given range and as a result produces some sound or noise is selected as a *feasible* action and therefore is also included in the planned course of actions.

The main responsibility of the BRI module rests upon the Plan Proposer, whose capabilities are at the same time supported by the ones of the Plan Proposer and Plan Executor. The Plan Proposer has to be described as the core of the BRI module.

7.4.3 The Plan Projector

Basically, the Plan Projector capabilities are grounded in understanding the effects of actions. Direct effects of actions require few understanding capabilities, and a strategy based on the direct matching between inputs and outputs of services suffices to implement the service *combination*[2]. The service combination capabilities are quite limited in comparison to the service composition ones, and whenever the sole matching of service inputs and outputs does not suffice to fulfill the requirements stated by the targeted behavioral response, it is necessary to resort to the higher capabilities provided by the service composition.

---

[1] Recall the *action-object* form of the request.

[2] Recall the difference between service composition and service combination

The advantages underlying service composition in comparison to service combination are mainly grounded in the management of the action effects, which in the former are far more sophisticated than in the latter. These complex effects involve the consideration of additional elements such as time, the context, the occurrence of concurrent actions, or even how context elements can be indirectly affected by actions that do not directly apply to them.

The success of the Plan Project is not only grounded in the management of the different types of effects, but in having identified and described those effects in the action descriptions that exist in the knowledge base. Both the reasoning capabilities and the knowledge base support are essential features for the success of the Plan Projector. It is useless for the Plan Projector to know about the different types of effects if those types have not been identified in any of the action descriptions. For example, it is indeed the Plan Projector's responsibility to understand the indirect effect of the *make noise* action as a means of making the person face the camera. However, it cannot be ignored that this indirect effect has to be somehow described in the knowledge base containing the action descriptions.

The Plan Projector role is twofold. On the one hand, it is in charge of determining all the effects caused by the actions proposed by the Plan Proposer. On the other hand, it is also responsible for devising complex effects of actions that can fulfill the sub-goals proposed by the Goal Detector at the request of the Plan Proposer. As has been described with the foregoing example, whenever basic service combination does not suffice to fulfill the goal state requirements, the problem is divided into sub-goals that are addressed as isolated problems. The solutions provided to each of these isolated problems are afterward concatenated into the comprehensive solution to the primary goal state.

These two functionalities of the Plan Projector also take part at two different stages of the behavioral response implementation task. Assessing whether the effects of actions fit the expected situation is a task that needs to be accomplished once at each of the stages that make up the action plan. Moreover, if the expected action effects do not lead the environment to the expected situation, the Goal Detector annotates the original goal state that was sought with the action proposed in the first draft, and this constitutes a sub-goal target for the Plan Projector that directly proceeds to the evaluation of complex effects, since direct ones have not succeeded in.

Algorithm 3 Plan Projector(plan, goalSituation)

```
stage = 0
while actions in plan do
    effects = getEffectsOf(action, object)
    if effects do not match subGoalSituation then
        get all actions $A = (a_1, a_2, ..., a_n)$ which complex effects leads to subGoalSituation
        while $a_i$ in $A$ and currentSituation differs from subGoalSituation do
            execute $a_i$ by the Plan Executor
        end while
        if currentSituation equals subGoalSituation then
            assert $a_i$ to the stage of plan.
        end if
    else
        send action to Plan Executor
    end if
    increment stage;
end while
Return Plan
```

Roughly, the Plan Project algorithm examines all the complex effects of actions that have been considered under the perspective of the common-sense knowledge. The algorithm should explore all the different types of effects of actions, providing special treatment to each of them, as described in the following paragraphs.

The analysis of those actions, whose *context-sensitive effects* are susceptible to leading to the intended sub-goal situation, reveal the set of constraints that need to be satisfied in order for the effects to take place. For example, recall that context-sensitive effect that states that the action of picking up a slippery object might result in the object being dropped if enough attention is not paid when picking it up. Therefore, the level of attention is the constraint that determines whether a slippery object is to end up being dropped or not. If the desired effect depends on any of these constraints, the next question is whether the constraint can be controlled, or it is out of the planner control, in which case it represents an uncertain aspect of the problem that cannot be affected by the planner. For example, a person can be warned about the need to pay attention when holding a certain slippery object, however, the level of attention that it is in fact paid by the holder cannot be controlled by the system. The attention level, therefore consists in one of those uncertain constraints that cannot be affected by the planner. On the contrary, there are some other context-sensitive effects that can be directly controlled by the planner. For example, think of the light conditions that determine the level of success of certain activities such as walking, reading, or cooking. If the light conditions can be directly controlled by the planner, using some sort of light sensors and actuators, then the effects of these tasks can be predicted. Poor or nil light conditions when performing the walking action result in stumbling situations, or walking into objects. There is therefore a difference between those constraints that can be directly controlled by the planner and those that are not.

Closely related to these context-sensitive effects that are outside planner control are the *non-deterministic effects* of events. These effects are subject to constraints that are out of the control and knowledge of the planner, and therefore, they cannot be determined beforehand. However, these non-deterministic effects are commonly constrained to a discrete space of possibilities. For example, stepping onto a wet floor might result in the person slipping or not. Despite the fact that the effects of these actions cannot be determined in advance, the set of possible effects are discrete and known beforehand. Knowing the possible effects of an action can be used by the planner whenever there is no any other action to accomplish. The action of flipping a coin is a paradigmatic example of non-deterministic effects. However, the resultant effects are constrained to be *heads* or *tails*. If the desired effect is that of obtaining a head, several attempts can be performed since probabilistically there is $1/(\text{state space size})$ percentage possibilities for that action to cause the desired effects.

Additionally, deterministic effects might also be affected whenever concurrent actions take place. These situations can lead to canceling, cumulative, or impossible effects. Canceling effects are helpful whenever the desired effects are exactly the contrary of the effects produced by a given action. For example, if the goal situation matches the contrary effects of the *door opening* action, the planner can easily question the knowledge base for those actions that cancel the *door opening* effects. Since both *open* and *close* have been described as actions with canceling effects, it can be easily inferred that the better way to achieve the aimed sub-goal is to perform the *closing door* action. *Cumulative* effects require more than one action to be performed at one stage. In this case, cumulative actions are programmed to take place at the same time instant, since otherwise effects do not produce the cumulative effects. For example, those actions that produce a measurable result are susceptible of producing cumulative effects, such as those involving sound or heat production. The cumulative effect takes place only when actions occur at the same time instant or time interval. The planner decides whether the result produced by the execution of a single action suffices to produce the desired effects. On the contrary, an appropriate approach to overcoming the shortage is by accumulating the effect of the same actions concurrently instantiated several times.

Finally, the *indirect effects* of actions is the last resource to resort to whenever any of the available actions do not directly produce the targeted sub-goal situation. In such a situation, it is worth analyzing whether any of the effects of the available actions can indirectly produce the sought effects. Recall the foregoing example presented in this section in which making a sudden noise causes an involuntary movement in a person to face the place from which the noise was originated. If the planner is capable of reproducing the cited conditions, and therefore it finally manages to make such an unexpected noise, as an indirect effect of the action the person will be facing the place in which the camera is located. In order to produce indirect effects it is necessary to undertake the action that indirectly causes the desired indirect effects.

It needs to be highlighted that the core functionalities of the proposed service composer are mainly grounded in this unit. Additionally, it can be noted that the service composition approaches presented to date only concern the direct effects of events, performing a simple matching of services inputs and outputs. The consideration of additional types of effects of actions enriches the service composer capabilities so as to provide service composition functionalities, rather than simple service combination.

7.4.4 The Plan Executor

The Plan Executor unit has already being introduced in the previous unit descriptions. This unit is instantiated by the Plan Projector unit whenever an action has been selected as part of a plan that leads to a target goal. This unit, rather than executing the whole plan outlined by the Plan Projector, is instantiated by the Plan Projector to execute a single pair of the form *action - object*.

The reason why the Plan Executor does not run a complete plan is the same, already mentioned above, the unpredictability of the real world interactions. Therefore, it is necessary to evaluate at each stage of the plan how the execution is evolving with regard to the expected sub-goal situations. Recall that each action composing the planned course of actions is intended to produce a sub-goal situation. After the execution of each of these actions, the resultant situation is evaluated in comparison with the expected one. If they match, the following action is executed, as stated by the Plan Projector. Whenever a deviation between the current situation and the sub-goal situation is detected, the Plan Projector gets engaged in determining the set of actions that finally produces the desired effects.

The main challenge that needs to be faced by this unit consists in determining how to automatically instantiate actions from the available services, managing the device heterogeneity that populates Ambient Intelligence environment. Once again, the implementation of the semantic model at the service level is what allows the Plan Executor to instantiate actions in a common and known way, independently of the underlying technology. Available services should implement a wrapping interface that provides a common access to all of them, based on the semantic model presented in Chapter 3.

The information about the services that provide a given set of actions is held in the knowledge base. Therefore, actions are associated to the services that provide them. Additionally, those services are also organized according to space restrictions. For example, if the action that *captures* a *snapshot* from a given room can be provided by a service of the type *camera service*, this action can only be performed by those services that are being provided at the same room location where the action is being requested. Common-sense knowledge about space and time has also been incorporated into action planning to discern between services that are available to those that are not.

The *feasibility* aspect of an action, which was referred to during the description of the Plan Proposer algorithm, does not only refer to the fact that the given action is being provided by a certain service. On the contrary, *action feasibility* involves several common-sense aspects such as existence, time, and space. Only when a service is accessible at a given time and location, to provide a required action, it is said to be a *feasible action*. Otherwise, the action falls into the group of *non-feasible* actions and therefore, it cannot be instantiated as part of the plan.

7.5 Interim conclusions

The conclusions that can be drawn from this chapter should be set together with the ones obtained from the previous chapter. Both chapters comprise the section that addresses the challenge of how to enable an Ambient Intelligence system to decide when and how to react to certain context situations, in an autonomous way. This complex task has been divided into two problems of a smaller entity, that of determining what is the most appropriate reaction to a given situation, and the problem of how to implement the outlined behavioral response. This chapter has been mainly addressed to the latter problem, as part of the Behavioral Response Implementation (BRI) module.

The analysis of the requirements that need to be fulfilled by a strategy intended to implement the behavioral response outlines provided by the Behavioral Response Generation (BRG) module, leads to the conclusion that a service composition approach is the most appropriate means. It has to be mentioned that the implementation of the behavioral response outlines has to be accomplished in an autonomous way, that is, without resorting to human directions. The simple analysis of the targeted situation should suffice to work out the course of actions that, from the current situation, drives the system to the goal state.

To the best of our knowledge, approaches presented to date to address the composition problem do not consider the complex effects of actions that can result in the most appropriate service composition, overcoming the poor results retrieved by those systems when the direct match between service inputs to outputs is not enough to address the composition request.

Part IV

Validation and discussions

Chapter 8

Validation Results

> **“A thinker sees his own actions as experiments and questions–as attempts to find out something. Success and failure are for him answers above all.”**
>
> *– Friedrich Nietzsche*

Summary – *This chapter concludes the part dedicated to validating this research. In order to do so, several case scenarios have been devised to encompass the different benchmark problems previously proposed for evaluating the common-sense capabilities of a system. These scenarios are described in the questionnaires that have been presented to a wide spectrum of social groups. These people have been asked to interpret the scenario descriptions and to evaluate the system’s interpretations.*

8.1 Introduction

This chapter focuses on evaluating the quality of the responses obtained from the prototype implementation described in the Appendix A. Nevertheless, the subjectivity and qualitativeness of evaluating the goodness of the responses makes its automation unfeasible. There are different factors involved in the evaluation that prevent the process from being automated, (i.e. preferences, likes, dislikes, fears, etc.). Given that the ultimate goal of this thesis is to emulate the human capability to respond to a great number of unexpected and unforeseen situations, the most appropriate way to assess the performance of the proposed solution is to contrast the system responses with those that would have been provided by humans.

In this sense, the approach followed consists in evaluating the response of the system in a set of paradigmatic case scenarios. The result obtained, given the representativeness of the selected scenarios, can be generalized into reasonable conclusions about the goodness of the proposed approach. It has to be noticed that these paradigmatic scenarios have been carefully designed in order to encompass all the different key issues of common sense described in previous chapters. Obviously, it is not possible to evaluate all the different situations that might take place in an open world. However, as stated by E.K. Mueller in [100], automating common-sense reasoning basically consists in devising appropriate means to addressed successfully such key issues.

Section 3.5 succinctly describes a set of benchmark problems, each of which individually brings to light the key issues of common sense postulated by E. Mueller in [100]. These benchmark problems need to be adapted in such a way that they can be provided to the system as though they were really taking place. The interpretation of those events as well as the reactions generated in response to such interpretations are compared and assessed by humans.

In this sense, the role of humans is twofold. On the one hand, respondents evaluate the different scenarios that have been especially designed and presented as a sequence of sensor measurements. For example, the scenario that describes room a in which the light has been turned on can be straightforwardly described by means of two statements. The first statement corresponds to the measurement before the light is switched on. In this case, the sensor measurement presents a value close to zero. The second statement signals the light change, presenting in this case a sensor measurement much greater than zero. However, it has to be noticed that different scenario descriptions might be consistent with the occurrence of these two events. For example, a scenario in which a flash of light occurs is also described by the same two statements. In this sense, there is not a unique relationship established between a sequence of events and its interpretation, since several interpretations might match up with the same sequence of events.

Respondents are therefore faced with different statements making up the description of the paradigmatic scenarios. The interpretations are collected and contrasted with the ones produced by the system. It is possible that system and respondents do not agree on the interpretation of a given scenario. The lack of consensus brings to light that further common-sense knowledge needs to be provided to the system so that it can reach the same interpretation. Moreover, it is possible that respondents themselves do not produce the same interpretation. In this case, a thorough quantitative analysis needs to be undertaken so as to analyze the different interpretations, and how these are related to the ones performed by the system.

Additionally, the second role played by humans in the experiment is that of having to grade their conformity with the interpretations made by the system. Despite differences in the interpretation that both the system and the respondent have provided, it is possible that the respondent considers the system interpretation as plausible. This is why in the first place respondents have been prompted to interpret the event sequences and then they have been prompted to grade the system's interpretations. Undertaking these two tests the other way around could possibly influence the respondent's opinion.

This chapter is organized as follows. First, Section **??** is devoted to proving the benefits of the composite services in contrast to basic or raw ones. In order to do so, an evaluation methodology is proposed and three case scenarios are provided to prove this assertion. The second section describes the evaluation platform. Third, the performed tests are described by enumerating the set of sensor values composing each case scenario that has been provided to the respondents. The next section presents the results obtained from the questionnaires, grouped according to different factors. Finally, the analysis of the data collected from the questionnaires leads to some important conclusions that are presented in the last section of this chapter.

8.2 Experimental validation

The design of an evaluation methodology is a crucial issue by which to assess the end user's degree of satisfaction, compare the performance of different alternatives and provide some feedback towards a process of continuous improvement and optimization. Nevertheless, this evaluation process entails a high degree of complexity since many different aspects, some of which are highly subjective, are involved. The methodology proposed here is outlined in Figure 8.1.

According to Figure 8.1, the process of evaluation for each event generates a vector of attributes, which are tightly related to the services generated and their characteristics. This vector is then evaluated to obtain the fitness (i.e. the goodness) of the system's response by means of a rule-based grading approach, which is generated using human expertise and the end user's expectations. Finally, the out-

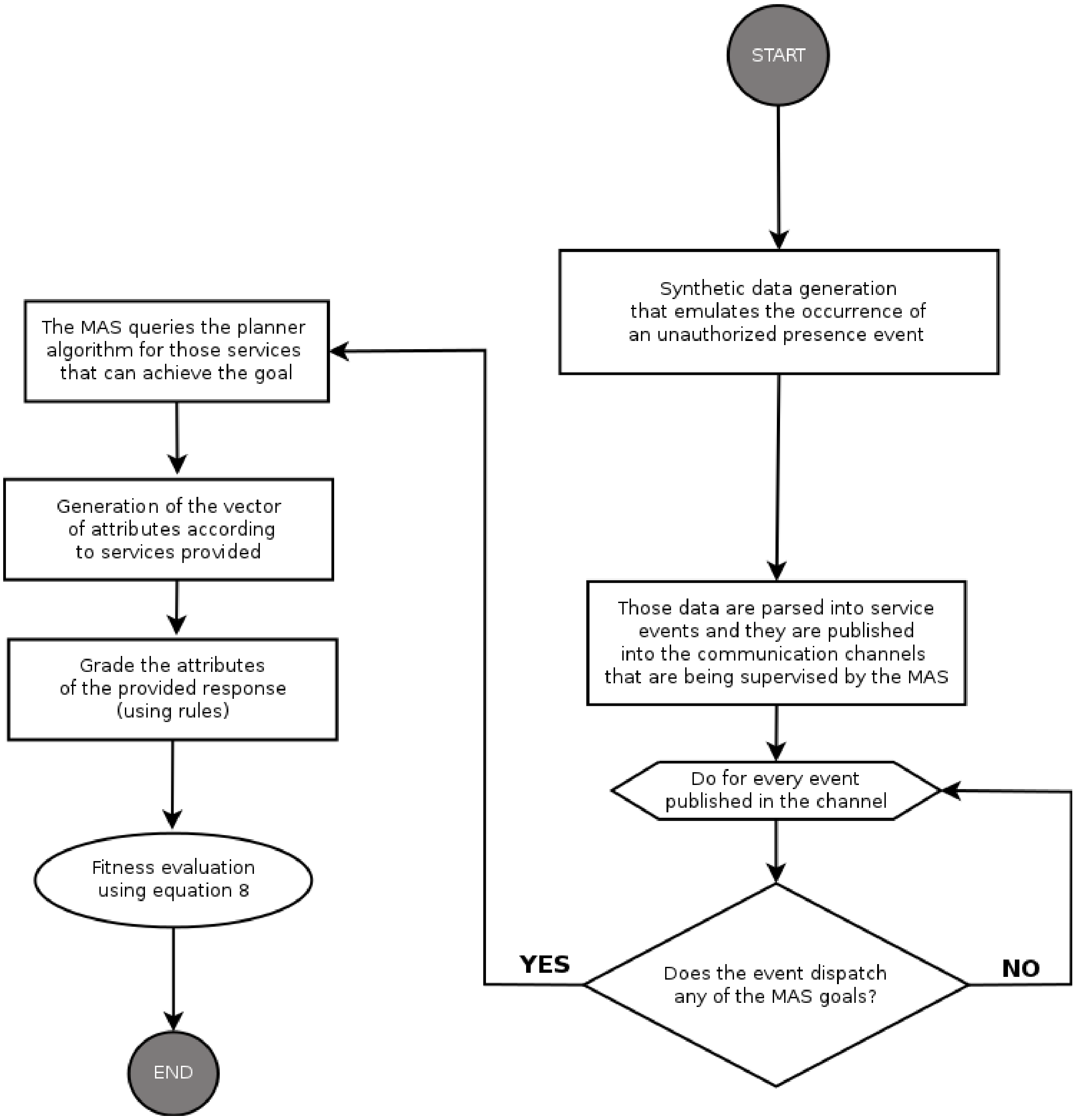


Figure 8.1: An evaluation process

come of the evaluation process provides the grading of the system's response and generates statistics and time series for a more in-depth analysis.

The evaluation process described here can be implemented to be executed on-line in real-time, or off-line from the data captured. Alternatively, it can be used at the design stage by implementing it in numerical simulations. The nature of the problem, which involves events and services, can be suitably addressed by means of discrete-event simulation tools (such as Arena, or Matlab/Simulink).

The result of the evaluation process may also be very useful for other purposes such as condition monitoring. Fitness variations can alert us to changes including device failures, vulnerabilities, environment and user habit variations that must be considered when redesigning the system.

The key elements of the evaluation process are described in the following subsections.

8.2.1 Vector of attributes

The vector of attributes must gather all the valuable information regarding the system response provided for a certain event. In the system presented it is particularly important to assess the benefits that composite services will bring. The vector of attributes could consist of the following elements: event start time; event duration; type of event; services provided, including the type of service (basic or composite), the number of basic or composite services provided, the service response times, and other particular characteristics.

8.2.2 Fitness evaluation

Once the vector of attributes has been generated for a certain event, the system response has to be evaluated. The approach adopted here is to create a set of rules designed to establish the criteria and quantify the fitness of the system response, according to the vector of attributes. Depending on the type of event, there are certain user expectations which must be fulfilled such as: response time, information provided by the system, and finally, how successfully the situation has been handled. The various aspects to be graded by the rules for each type of event are: number of basic services provided; the number of composite services provided; the ratio between composite services and basic services; the response time for each service; the usefulness of the services provided according to the type of event; the rate of success of certain services.

The grades given, based on the rules, must be properly weighted, according to the significance of each graded feature and added to obtain an overall fitness value.

8.2.3 An example

A simplified example is presented here to illustrate the evaluation process described. The example is a surveillance application, where the scenario considered is a room containing a presence sensor and a camera. The image of the camera can be processed by a face recognition software application.

In this example, only one type of event is considered: human presence in the room. With regard to the services provided, those which are most basic are the state of the presence sensor, the video streaming from the camera and the face recognition output; while the composite services are combinations of these according to the common-sense reasoning system implemented for automatic service composition. In this case, two composite services have been considered. One of them notifies the security staff when the sensor detects presence and automatically provides video images. The other composite service also launches the face recognition application and provides its output.

For each event, a vector of attributes is generated, which contains the most relevant information about the system response and the services provided. The fitness evaluation is then performed by grading each individual attribute according to a set of rules. Finally, the overall fitness of the system response is obtained by multiplying the grades of each attribute by a weighting and adding them all, as in Equation 8.1. The weight of each concept is established according to its contribution and how significant it is when assessing the goodness of the response. Attributes related to composite services have a higher weight than basic ones since they better fulfill user needs and will reduce human intervention when responding to events. Hence, the goodness of a response will be enhanced by composite services and the resulting fitness value will be higher.

$$F = \frac{\sum_{n=1}^{N} w_n \cdot g_n}{\sum_{n=1}^{N} w_n} .100 \qquad g_n \varepsilon [0,1] \tag{8.1}$$

Table 8.1: Simulation results

Case	**Mean of F**	**Std dev of F**	**% of Comp Services**
1	12.55	1.937	0
2	42.31	20.33	75.4
3	77.21	35.93	75.4

F is the fitness and N is the total of number of attributes, and w_n and g_n are the weight and grade of the nth attribute respectively. The value of F is normalized in the range 0 to 100. The example described has been simulated using probabilistic distributions to model the availability, rate of success and response time of the services previously described. Three different cases have been considered: in case 1 only basic services are provided; in case 2 basic and composite services are provided, and the face recognition application has a success rate of 10%; in case 3 basic and composite services are also provided, but the face recognition application has a success rate of 90%.

A simulation with 1000 events has been run for each case. Figure 8.2 shows the results obtained for the three different cases.

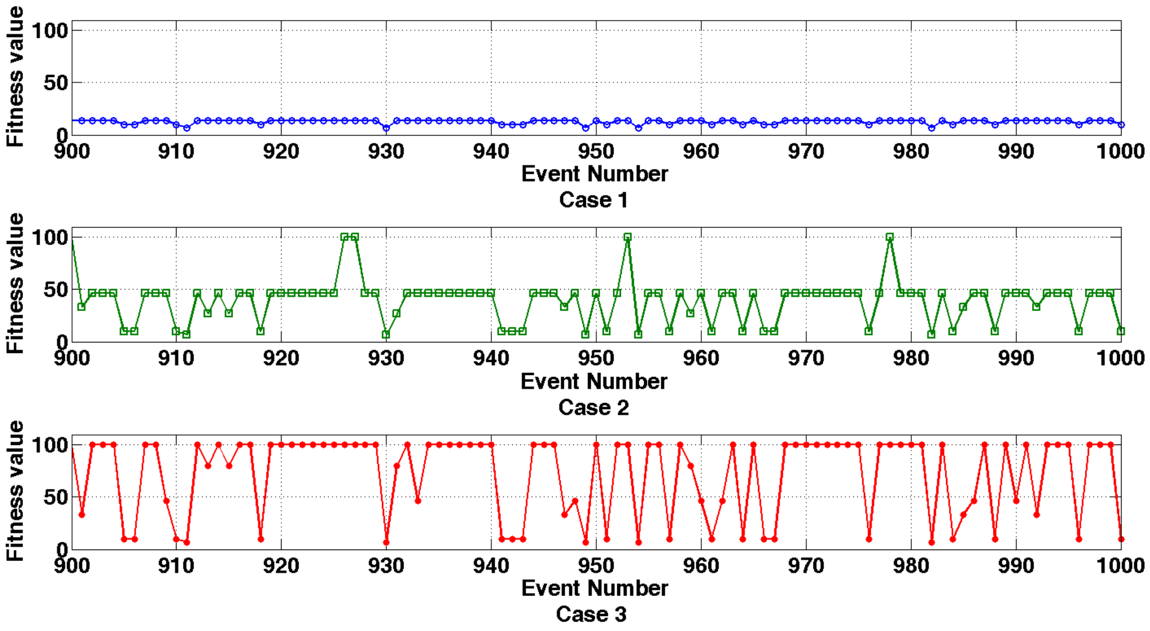


Figure 8.2: Simulation results

Table 1 shows the mean value and standard deviation of the fitness for each case. It also shows the percentage of events in which composite services are provided. It can be observed that case 3 gives the highest fitness in most of the cases, since it involves composite services that better fulfill the user's needs and the success rate of the face recognition application is higher.

It can be concluded from the observed results that case 3 gives the highest fitness in most of cases, since it involves composite services that better fulfill the user's needs and the success rate of the face recognition application is higher. The suitability of the evaluation methodology proposed here has also been proved to serve as a means to rate the goodness of system responses.

8.3 Test Description

The prototyped version of the architecture proposed in this thesis, described in the Appendix A, is used to run a set of benchmark problems. Additionally, a synthetic generator[1] of the events has been incorporated into the prototype in order to generate the appropriate event sequences that emulate the occurrence of the designed benchmark scenarios.

Table 8.2 presents the simulation settings. The first column lists the nine scenarios, the second column states the set of sensors used to describe each of the different scenarios and, finally, the third column lists the statements that characterize the different scenarios. These events have been synthetically generated and represented using a more human-readable representation. There is no difference between using synthetic or real events since both of them are parsed at the middleware layer to be stamped with the date and time at which the event takes place, the name of the sensor that generates the event, and the value it gives.

The way events have been described in Table 8.2 consists in an abbreviated and language-independent representation, used for the sake of the respondent's understanding. Rather than providing respondents with Scone-like sentences, it is more appropriate to abstract the scenario description from any particular representation language. However, in order to provide the reader with a general feeling of how real representation sentences look, the following code listing is provided. It consists of a brief extract of the code employed to describe the knowledge referred by a presence sensor:

```
(new-type {device} {thing})
(new-type {sensorDevice} {device})
(new-type {presence sensor} {sensorDevice})
(new-type {presenceSensorService} {service})
(new-indv {presenceSensorIndv} {presence sensor})
(new-indv {presenceSensorIndvService} {service})
```

Those statements starting with `new-indv` assert specific domain knowledge in the knowledge-base. For example, the `presenceSensorIndv` asserts the existence of a new individual in the domain. This inherits from its ancestor, the `presence sensor` type, all the properties and roles associated with it.

```
(new-indv {SimpleCapturer:default -p 12000} {presenceSensorService})

(the-x-of-y-is-z {offered-service}  {presenceSensorIndv}
   {SimpleCapturer:default -p 12000})

(the-x-of-y-is-z {provider-device} {SimpleCapturer:default -p 12000}
   {presenceSensorIndv})

(the-x-of-y-is-z {performs-action}  {SimpleCapturer:default -p 12000}
   {presenceDetection})

(the-x-of-y-is-z {agent-of} {presenceDetection}
   {SimpleCapturer:default -p 12000})
```

It should be highlighted that services, as in the case of the `SimpleCapturer:default -p 12000`,

---

[1] The Synthetic Data Generator is available at: http://ailab.eecs.wsu.edu/casas/datasets.html

are provided to the knowledge-base with the name of their *proxy*[2]. This coupling between the middleware and the knowledge-base system should not be considered negatively since, as depicted in Figure A.4, the proxy address is what allows the Multi-Agent System to automatically instantiate services directly from the sequence of actions returned by the planning algorithm.

Two types of knowledge should be considered in the knowledge-base, static and dynamic. The sort of knowledge known as static refers to the knowledge description that describes how the world works, the so-called common-sense knowledge. On the other hand, the dynamic knowledge refers to the specific domain knowledge that can be stated and eliminated from the knowledge-base, depending on whether or not it exists in the domain under consideration. In this sense, whenever a new sensor appears on the scene, the Multi-Agent System is in charge of introducing a new individual into the knowledge-base. This task can be accomplished in an unsupervised manner because the inheritance mechanisms relieve the agent of having to provide all the properties that characterize a type of sensor already described. In this sense, introducing a new individual sensor of type `presence sensor` means that this new individual inherits all the properties associated with the *typical* presence sensor.

The events used to describe each of the nine benchmark scenarios are automatically introduced into the knowledge-base, becoming part of the dynamic knowledge it holds. The events produced by new sensor measurements, at specific time-instants, are introduced as statements of the following relation:

```
(new-relation {captured-measure-at-time-instant}
        :a-inst-of {sensor}
        :b-inst-of {measure}
        :c-inst-of {time-instant})

(new-statement {presenceSensorIndv} {on} {2011-05-02 10:59})
```

The last statement is used to state that a certain presence sensor has noticed a moving object at the specified time.

The Multi-Agent System is in charge of capturing the sensor measurements published in the communication channels, parse them into the Scone representation format, and then introduce them into the knowledge-base. Due to the fact that the Multi-Agent System works as an intermediary between the environment and the knowledge-base, it is also responsible for detecting those abnormal situations that require further analysis, on the basis of the goals pursued.

8.3.1 Questionnaire development

It is known that there exists a strong connection between common sense and human characteristics associated with socio-cultural, economic, gender, or age factors, among the most relevant. For this reason, the task of gathering common-sense knowledge has to be undertaken bearing in mind that such knowledge can be influenced by these factors.

Knowledge-base systems such as Cyc or OpenMind are fed with information provided by thousands of volunteers. Sometimes, the knowledge gathering process is incorporated into the use of games, or some other appealing mechanisms that relieve the volunteers of having to make use of formal representation language. In this sense, the task is completely automated and does not require the volunteers to have any other specific ability but that of holding common-sense knowledge. Despite the fact that the collected knowledge has to be supervised before being definitively introduced into

---

[2] The *proxy* concept is used to refer to the address in which clients can find a given service. This concept has been explained in more detail in Section A.2.

Table 8.2: Simulation Configuration

Scenario	Sensors	Events
Scenario 1	lamp-sensor (on/off)	2011-05-02 09:50 lamp-sensor on
Scenario 2	oven-sensor (on/off) fire-sensor (on/off) thermostat (18-51 Celsius)	2011-05-02 13:29 thermostat 21 C 2011-05-02 14:00 oven-sensor on 2011-05-02 14:05 fire-sensor on 2011-05-02 14:25 thermostat 24 C 2011-05-02 14:58 thermostat 26 C
Scenario 3	liquid-on-floor-sensor (on/off) presence-sensor (on/off)	2011-05-02 16:05 liquid-on-floor-sensor on 2011-05-02 16:07 presence-sensor on
Scenario 4	tap-sensor (open/closed) stopper-sink-sensor (in/out)	2011-05-02 08:04 tap-sensor closed 2011-05-02 10:59 stopper-sink-sensor put-in 2011-05-02 11:04 tap-sensor open 2011-05-02 11:05 tap-sensor closed 2011-05-02 11:15 tap-sensor open 2011-05-02 16:05 tap-sensor closed
Scenario 5	door-sensor (open/close) Door-activator (on/off)	2011-05-01T20:30 door-sensor closed 2011-05-01T20:31 door-activator on 2011-05-01T20:31 door-sensor closed
Scenario 6	thermostat (18-51 Celsius) oven-sensor (on/off) fire1-sensor (on/off) fire2-sensor (on/off) fire3-sensor (on/off) fire4-sensor (on/off)	2011-05-02 12:16 thermostat 21 C 2011-05-03 14:41 oven-sensor on 2011-05-03 14:42 fire1-sensor on 2011-05-03 14:43 fire4-sensor on 2011-05-03 14:45 thermostat 23 C 2011-05-03 14:55 fire2-sensor on 2011-05-03 14:59 fire3-sensor on 2011-05-03 15:39 thermostat 33 C 2011-05-03 15:50 fire2-sensor off 2011-05-03 15:57 thermostat 31 C 2011-05-03 15:52 oven-sensor off 2011-05-03 15:58 fire3-sensor off 2011-05-03 16:05 fire4-sensor off 2011-05-03 16:09 fire1-sensor off 2011-05-03 17:05 thermostat 21 C

Scenario 7	kitchen-presence-sensor (on/off) kitchen-lamp-sensor (on/off) bedroom-presence-sensor (on/off)	2011-05-02 10:59 kitchen-presence-sensor on 2011-05-02 10:59 kitchen-lamp-sensor on 2011-05-02 10:59 bedroom-presence-sensor on
Scenario 8	presence-sensor (on/off) lamp-sensor (on/off) light-sensor (0-100)	2011-05-02 10:59 presence-sensor on 2011-05-02 10:59 lamp-sensor on 2011-05-02 11:01 light-sensor 80 2011-05-02 11:31 light-sensor 20 2011-05-02 11:32 presence-sensor off 2011-05-02 12:37 presence-sensor on 2011-05-02 12:37 light-sensor 81 2011-05-02 12:38 presence-sensor off 2011-05-02 23:59 presence-sensor on 2011-05-03 00:00 presence-sensor off 2011-05-03 00:01 light-sensor 70 2011-05-03 03:51 light-sensor 2
Scenario 9	door-sensor (open/closed) kitchen-presence-sensor(on/off) kitchen-light-sensor (0-100)	2011-05-01 20:31 door-sensor closed 2011-05-01 20:31 kitchen-presence-sensor on 2011-05-01 20:31 kitchen-light-sensor on 2011-05-01 20:31 light-sensor 80 2011-05-01 20:36 light-sensor 0 2011-05-01 20:36 light-sensor 10

the knowledge-base, it saves time in the gathering process, and what is more important, it benefits from the diversity of perception of the different people involved in the process.

One of the main shortcomings of this research is that the majority of the common-sense knowledge used for the prototype implementation has been introduced exclusively by the author of this thesis. For that reason, the validation questionnaires are also intended to detect and overcome the possible partiality of the knowledge representation.

The experiment depicted here therefore pursues a twofold aim in questioning people. On the one hand, people have been prompted to interpret a sequence of sensor measurements as though they had been generated by the occurrence of a certain known situation. On the other hand, after having interpreted those event sequences, people are requested to grade the solution provided by the system to the same event sequences. In this sense, the production of the questionnaire has been undertaken by capturing the events that have been synthetically triggered when emulating each of the benchmark problem scenarios.

Table 8.2 lists in its third column the set of event sequences or sensor measures that characterize the different benchmark scenarios. It can be seen that each scenario has been run in isolation so as not to distract attention from the key issue being analyzed. For example, Scenario 3 is only devoted to analyzing whether the non-deterministic effects of events are being appropriately addressed. Nevertheless, the mental state key issue is implicit in the same scenario, due to the fact that the plausibility of potential harm associated with a slippery floor deflects the system from its desired state of "preventing human integrity from being compromised". Similarly, the default reasoning key issue is also considered in conjunction with the effects of events, in Scenario 1. The common-sense law of inertia is considered in Scenario 4, along with the delayed effects of events. Finally, the continuous change has been included in Scenario 6, in which the temperature increase is associated with the activity of cooking, and it is also expected to be continuously increasing until affected by an external event.

Each scenario is translated into a set of time stamped events. These synthetic events are provided to the system as a text file, from which the events are gathered and parsed into event notifications (date and time, sensor identity, and the modified state or value) that are published through the middleware communication channel. The intelligent agent subscribed to that communication channel detects whenever a new event is published. As a result of a new notification, the intelligent agent is responsible for introducing the parsed events into the Scone Knowledge-Base system.

Note that besides the parsed events, the Scone knowledge-base system has been provided with a description of the environment in which the different scenarios take place. For example, the kitchen is said to be on the first floor, with no windows, one entry door, an electric cooker, and so on. That information has been introduced before running these tests, since many of the interpretations and inferences are context-sensitive.

Apart from introducing events into the Scone knowledge-base, the intelligent agent evaluates each of these events from the point of view of the beliefs and goals that it holds. The purpose of this evaluation is to detect any goal deviation or any situation that leads the environment to an undesired state.

The following items describe each of the benchmark scenarios along with the expected responses and interpretations of the prototype version of the system.

- **Scenario 1.** This scenario has been devised to assess the performance of the systems when it comes to representing and evaluating the effects of events. In order to do so, the system has been prompted to list the effects of the *turn lamp on* event. The resulting effects are: a) The lamp device is on; b) The place in which the lamp is located is now illuminated.

- **Scenario 2.** This scenario is designed to evaluating the system capability to represent and reason about event effects that are context-sensitive. In this situation, the room temperature is a *fluent* that is likely to be connected to continuous actions whose effects cannot be directly known. The intelligent agent considers temperature as a relevant fluent to be monitored and therefore, the rapid increase of temperature[3] triggers the suspicion of a potentially abnormal situation going on. This abnormal situation creates the need for the system to determine the underlying ongoing situation. In order to do so, the system analyzes those sources that might have affected the kitchen temperature. And especially, those that are responsible for producing heat (or a temperature increase) are looked at. Among the retrieved devices (the heating system, the lighter, the candle, the oven, or the kitchen fire, among others) the ones that are available in that particular kitchen and are also turned on are the oven and the cooing hobs. Due to the fact that temperature does not exceed 45 Celsius, the activity that involves the kitchen rings and the oven is interpreted as that of cooking. The interpreted situation is therefore considered to be normal.

- **Scenario 3.** This scenario is intended to evaluate the system's ability to represent and reason about non-deterministic effects of events. The event that activates the `liquid-on-floor-sensor` is indirectly stating that the floor is wet and therefore slippery. Due to the fact that the house has been described as having a single inhabitant, with no pet, the activations of the presence sensor are interpreted as though a person were walking near the presence sensor. This situation is labeled as abnormal since the person walking activity requires a solid floor to step on (as has been stated in the `thorough context` of the `person walking` event description). Since the floor which the person is likely to step on is slippery, the situation is considered abnormal. In this case, the intelligent agent triggers a plan that leads the system to maintain the goal of preventing human integrity from being compromised. The abstract plan associated with this situation is intended to notify the walker. In order to do so, and given that the kitchen is equipped with a sound device, the intelligent agent simply emits a message alerting to the abnormal condition of the floor.

- **Scenario 4.** This scenario has been designed to assess the system's capability to foresee the delayed effects of events. The event of opening the tap has the effect of water running through it to a recipient or an open space. The recipient type determines the final effect of an event. In this case, if the sink has the stopper in, the water does not run through the pipe and the water contained in the sink starts increasing its level. After noting this event, the intelligent agent identifies a potential abnormal situation. After a period of time has elapsed, and given that the water level has increased as a result of time and the sink size, the intelligent agent determines that the water is likely to spill onto the floor. Therefore, the situation is considered abnormal, and the associated plan of notifying abnormal situations, once again, is instantiated as a message through the kitchen sound device.

- **Scenario 5.** This scenario has been devised to evaluate the system's capability for dealing with cancelling effects of events. In this scenario, the `open door` action has been described as an action whose effects can be cancelled by any other action that applies the same moment in the opposite direction. In this sense, the action intended to open the door results in a different effect than the one expected, since the door remains closed. This mismatch between the expected

---

[3] It is considered a rapid increase since this fluent varies at a maximum rate of 3 degrees Celsius per hour. Out of this range, the situation is considered abnormal.

effects and the sensed ones leads the intelligent agent to try to determine the *primary cause* for that abnormal situation, before any other plan can be undertaken. The primary cause for the door to remain open after activating the activator is that an equal or greater moment is being applied to the door in the opposite direction.

- **Scenario 6.** This scenario is similar to the one described in Scenario 2, with the only difference that it considers that there is a sensor for each of the cooker hobs. The purpose of this scenario is to determine the system's capability for dealing with the cumulative effects of events. Given the same situation as the one in Scenario 2, the system reaches the same conclusion. It should be noted that the kitchen hobs have been modeled as a set of devices with the capacity to transform electrical energy into heat, providing a variable power that ranges from zero to the maximum value they can support. In this sense, the temperature increase cannot exceed the threshold associated with the consumed energy. Since, in this situation the increase does not exceed that threshold, the situation is therefore considered a normal cooking activity.

- **Scenario 7.** This scenario has been devised to evaluate the system's ability to represent and reason about concurrent events. It is in the nature of humans that our spatial location can only be associated with a single position at a time. Therefore, this scenario presents two events in which, in absolute coordinate positions, the person is located in two different places at the same time. Note that the house has been considered to be occupied by a single person, therefore, given that a person cannot be in two distant positions at the same time instant, an abnormal situation is going on. Once again, the intelligent agent is prompted to determine the most appropriate plan to handle this situation, apart from the simple notification to the house occupant.

- **Scenario 8.** This scenario is intended to assess the system's ability to deal with the indirect effects of events. This scenario depicts a kitchen with no windows, in which the brightness level is monitored by a sensor device. The brightness level is considered to be an additional fluent, as is the temperature. This means that the variation suffered by this fluent can provide the system with important information about the ongoing situation. The sudden decrease in the brightness level is interpreted as a possible abnormal situation that needs to be analyzed, since it has not been preceded by the lamp sensor deactivation, which implies that the person has not switched the lamp off. When queried about the events that might lead the context to a blackout situation, it points out that it is either due to an electrical failure or to the kitchen light having burned out. Since it is not the first case, the system concludes that the primary cause for that situation is the kitchen light burning out.

- **Scenario 9.** This scenario is designed to assess the system's capability to deal with spatial reasoning. In this scenario the kitchen has been depicted as a closed room, with no windows and a single entry point. In this sense, given a blackout situation, the new brightness value represents an abnormal situation since it is expected to be completely dark. The system searches for a source of brightness, in the same way the system searched in Scenario 2 for sources of heat. It reaches the conclusion that there is a set of devices that are capable of providing that level of brightness, such as a lighter, candle, or an opened fridge, among others. None of the actions that could cause such a brightness level are considered to compromise the human or house integrity and therefore the situation is considered to be normal.

8.4 Test Results

The test described test has been carried out in two different and separte stages that correspond to the two questionnaires that respondents have been asked to answer.

1. **First questionnaire:** This questionnaire is intended to answer the following question: *How do people interpret and react to the scenarios described?* In this sense, the data in Table 8.2 is provided to people in a more friendly and human-readable manner. Additionally, people are also provided with some relevant information about the environment or the significance of the sensors/actuators because not all the respondents are familiar with the different types of sensor devices. Respondents are asked to interpret the events and to provide feasible solutions to those scenarios whose interpretation requires a system intervention.

2. **Second questionnaire:** This questionnaire is designed to assess the system performance, from the point of view of its ability to address common-sense reasoning and knowledge management. This questionnaire is also intended to determine whether the interpretation and solutions provided by the system are in consonance with what the majority of the respondents think as *common-sense interpretations and responses*. In order to do this, respondents assign a grade according to their level of conformity with the system performance.

The first questionnaire has been anonymously answered by 46 people and the second only by 35. Table 8.3 and 8.4 summarize the personal information provided by the respondents. It should be noted that some of them have decided not to provide the personal information.

- **Group A**: Younger than 15 years old.
- **Group B**: Older than 15 and younger than 25 years old.
- **Group C**: Older than 25 and younger than 65 years old.
- **Group D**: Older than 65 years old.

Regarding the academic background, the following groups have been considered:

- **Group N**: For those that have not completed any academic studies.
- **Group P**: For those that have completed primary education.
- **Group S**: For those that have completed secondary education.
- **Group U**: For those holding a university degree.

Table 8.5 summarizes the different interpretations provided by the respondents, along with the number of people who have agreed on each of the different answers. Those answers that are not relevant have been grouped in the "*Other*" answer. Recall that respondents are provided with a brief introduction to the context in which the events take place, and they are supposed to interpret the event sequence listed in the third column of Table 8.2.

The results retrieved from the second questionnaire are listed in Table 8.6. Recall that the purpose of this questionnaire is to assess the system performance from the point of view of how satisfied the respondents are with the interpretations and the solution provided by the system. Respondents have

Table 8.3: Personal information of respondents to the first questionnaire

Academic Background		Age		Gender		Nationality	
Group	**Number**	**Group**	**Number**	**Group**	**Number**	**Group**	**Number**
N	-	A	2	Male	25	Spanish	36
P	1	B	15	Female	13	German	1
S	10	C	18			Colombian	1
U	27	D	-				

Table 8.4: Personal information of respondents to the second questionnaire

Academic Background		Age		Gender		Nationality	
Group	**Number**	**Group**	**Number**	**Group**	**Number**	**Group**	**Number**
N	-	A	2	Male	24	Spanish	32
P	1	B	6	Female	13	German	1
S	8	C	27			Colombian	1
U	26	D	-			Mexican	1

been provided with the scenario descriptions set out in Section 8.3.1. Table 8.6 summarizes the grades obtained.

From the results obtained, and with regard to the academic background, it seems that those from the groups that do not hold a university degree tend to be more satisfied with the system responses and interpretations than those who do hold one.

From the overall results, Scenario 5 seems to be the one with which the people questioned are least satisfied. Additionally, the results obtained from Scenarios 3 and 8 are, in general, less positive than those obtained from the other scenarios.

An additional drawback of the results is that mostly Spanish (and only Spanish-speaking) people have been questioned, and therefore, it has not been possible to evaluate how different socio-cultural backgrounds affect the responses obtained.

8.5 Data Analysis

The previous section has been basically devoted to transcribing the raw data collected from the two questionnaires undertaken by a heterogeneous group of respondents. This section is intended to analyze those data from a quantitative and qualitative point of view, in order to establish direct connections to the research questions posed at the beginning of this thesis.

The reason why Ambient Intelligence scenarios, as described in [35], have not yet become a reality is the main research question that motivates this thesis. In this regard, several working hypotheses were postulated in Chapter 1, as summarized here:

- Reasoning capabilities need to be more carefully developed.
- Cognitive and planning capabilities inevitably depend on holding vast amounts of commonsense knowledge.
- Ambient Intelligence systems should be grounded in a philosophical theory of actions, that provides a thorough understanding of how they should be modeled.

Table 8.5: Interpretation results

Scenario	Interpretations	Number of people
Scenario 1	Light has been switched on at 9:50, May 2	10
	The kitchen is illuminated	11
	The sensor is activated only because the light is on	2
	Ambiguous description. Events cannot be interpreted	1
	Someone has entered the kitchen and turned on the light	5
	Only if the light works properly the kitchen is illuminated	1
	Since it is daylight, kitchen illumination only increases a little	1
	Others	5
Scenario 2	Someone is cooking lunch	5
	Temperature increase is due to the cooker and oven heat	26
	The two previous interpretations	5
	Other	5
Scenario 3	There is liquid is on floor and a person enters the room	9
	A person enters the room to clean the floor	9
	Boiling water spills out of the pan	2
	Slippery floor	7
	The liquid has not been dropped by a person	3
	Other	4
Scenario 4	Nothing relevant happens	5
	The sink starts filling up	5
	An overflow occurs	23
	Someone forgot to turn the tap off	1
	Other	1
Scenario 5	Erroneous sensor	8
	The door opens and closes normally	15
	Something is blocking the door	5
	Other	6
Scenario 6	Cooker hobs are being turned on and off	2
	The more cooker hobs being used the higher the temperature.	10
	Somebody is cooking	4
	Kitchen hobs 1 and 4 are bigger than the rest	2
	Other	5

Scenario	Interpretations	Number of people
Scenario 7	A person cannot be in two different places at the same time	4
	Nothing relevant apart from the events described	10
	There is someone else in the house	8
	Faulty sensors	7
	The two previous interpretations	3
	Some movement in the kitchen activates the sensor (toaster)	3
	Other	3
Scenario 8	It is due to a shadow or occlusion problem	4
	Light intensity is regulated by the person's presence	8
	Nothing relevant apart from the described events	6
	Light bulb burnt out	4
	Light intensity changes are due to natural light	2
	Other	2
Scenario 9	The light bulb burns out and then a candle or a match is lit	8
	Occlusions or shadows affect the sensor measurement	4
	Faulty sensor	5
	Light intensity is regulated by the person's presence	7
	Nothing relevant apart from the events described	7
	Other	2

The first step in analyzing the findings is to transform the raw data into a more sophisticated format that facilitates its analysis. This task has been undertaken in Tables 8.3, 8.4, 8.5, and 8.6 which summarize and show the questionnaire information in a more condensed format. Next, once the data has been reduced, the focus is on identifying patterns in the responses in such a way that a more elaborate analysis of the pattern can lead to more interesting findings.

It is a common pattern that respondents tend to generalize well when it comes to identifying high level activities, such as cooking, out of a sequence of sensor values. For example, the event sequence of Scenario 3 is interpreted as liquid spilling, and a person entering the kitchen to clean up the floor. An additional pattern can be identified in the major agreements that characterize those responses that involve rational explanations to sensor measurement. For example, in Scenario 2, the temperature increase is due to the cooking activity. In other words, the temperature increase is considered irrelevant whereas the majority of the respondents focus at identifying the causal explanation for that increase. It is also worth noticing the sort of expectations given by respondents when it comes to explaining *a priori* unfeasible measurements. It is quite common to think that these measurements are the result of a faulty sensor, rather than looking for a more rational explanation. For example, in Scenario 7, eight respondents argue that there must be someone else in the house, whereas seven other respondents state that these measurements are due to a faulty sensor.

It is also possible to find some relevant patterns in the answers provided by each respondent, in terms of the group of people into which each of them falls. Two of the respondents are females and younger than 15 years old. These two girls have answered to the majority of the questions that they did not know how to interpret the events or that they did not understand the event sequence. In contrast to adults, none of whom have shown any difficulty in evaluating the scenarios, children do not

Table 8.6: Test results

Scenario	Answers	Number of people	Scenario	Answers	Number of people
Scenario 1	Very Good	21	**Scenario 6**	Very Good	18
	Good	7		Good	12
	Fair	5		Fair	3
	Bad	1		Bad	0
	Very Bad	0		Very Bad	0
Scenario 2	Very Good	21	**Scenario 7**	Very Good	19
	Good	9		Good	11
	Fair	3		Fair	4
	Bad	1		Bad	0
	Very Bad	0		Very Bad	0
Scenario 3	Very Good	16	**Scenario 8**	Very Good	11
	Good	9		Good	15
	Fair	8		Fair	7
	Bad	1		Bad	1
	Very Bad	0		Very Bad	0
Scenario 4	Very Good	18	**Scenario 9**	Very Good	12
	Good	13		Good	11
	Fair	3		Fair	4
	Bad	0		Bad	0
	Very Bad	0		Very Bad	0
Scenario 5	Very Good	12			
	Good	11			
	Fair	9			
	Bad	1			
	Very Bad	0			

understand the terms used to describe them. Additionally, respondents older than 50 and who do not hold a university degree, tend to expects sensors to do something more than just take measurements. For example, it has been a common pattern regarding the light sensor that, rather than measuring the illumination level, this sensor is responsible for regulating it. It makes sense to them that wherever a sensor is deployed, extended capabilities should be provided to the environment.

Once the collected data have been collated and analyzed for pattern identification, it is now possible to evaluate how the filtered data are related to the working hypotheses and the aforementioned key research questions. In this sense, it should be highlighted that both the prototype implementation and the questionnaires make up the context of the specific sample in which the research questions are to be answered and the working hypotheses are to be tested.

To the question of why Ambient Intelligence environments have not yet become a reality, the prototype implementation brings to light very important conclusions. Instead of building the complete system theorized in this thesis, the construction of a prototype version is a that is supported by one of the main claims postulated here: only by leveraging common-sense knowledge can Ambient Intelligence systems be permitted to understand and reason about the context, in a non-supervised manner. Due to the unfeasibility of collecting all the common-sense knowledge held by humans, a reduced sample can be used to run the test whose results can afterward be generalized. The sample context is that of Ambient Assisted Living, such as a house, and only events and actions that can take place there have been represented in the knowledge-base system. In order to explore the most paradigmatic situations of common-sense reasoning, a set of benchmark problems has been carefully designed. After running the test, and collecting the system responses, these are provided to the questionnaire respondents. They have to assess the system performance in comparison with what a person would have done under the same circumstances. A percentage between 30% to 60% of the respondents consider the system's understandings and reactions to be very good, and between 20% to 50% of the respondents consider them good. This means that around 80% of the questioned people consider that the system responses are good or very good.

The prototype implementation is supported in a knowledge-base system that has only been provided with information about the most common human activities carried out in a domestic environment, and the sensor and actuator features and services. Using this sample context, respondents have positively evaluated the system's interpretations and reactions, what therefore supports the initial hypothesis of common-sense knowledge playing a key role in enabling Ambient Intelligence.

Scenarios 5 and 9 have obtained the lowest grades,providing a hint as to the most challenging issues of common-sense reasoning, which are the simultaneous effects of events and spatio-temporal reasoning. This finding supports the hypothesis that poor reasoning mechanisms do not suffice to address the challenges involved in dealing with Ambient Intelligence environments. This result shows that simultaneous events and spatio-temporal reasoning need to be more specifically addressed so as to improve the understanding and responses provided by the system when these issues are involved.

A thorough interpretation of the data also leads to the conclusion that those responses that involve causal explanations achieve a higher level of agreement among the different respondents. This important finding supports the initial hypothesis that a philosophical theory of action needs to be adopted so as to automate the understanding and reasoning process involving actions. The Davidsonian theory of *primary reasons* described in Section 4.4 advocates the existence of a primary reason or a cause that motivates the realization of an action. Similarly, knowing the effects of an action can also lead to discovering the primary reason or the cause for that action to take place. In this sense, provided that temperature is increasing in the kitchen and that turning on the cooker produces heat, it can be causally explained that a cooking activity is being carried out. Producing causal explanations is therefore one of the main mechanisms used by humans to understand the situations taking place in the context. This finding therefore supports the working hypothesis that advocates the implementation of an action theory, as described in Chapter 4, by means of a semantic model of actions and events.

The findings of this data analysis process have not only been useful to support the working hypothesis postulated at the beginning of this research, but also, they have proved the feasibility of achieving Ambient Intelligence as envisaged in [35], by leveraging common-sense knowledge and reasoning capabilities.

8.6 Conclusions

This work has been mainly devoted to justifying the important role that common sense plays in achieving Ambient Intelligence systems that are indeed intelligent. Inspired by the work of E. Mueller [100], the key issues of common sense listed by this author have been used as a requirement checklist for leveraging common-sense in systems for Ambient Intelligence. Once these items have been adapted into an architectural prototype for Ambient Intelligence[4], the next stage consists in validating the proposed approach.

The most common approach to testing this these sort of system is based on running a set of benchmark problems, especially designed to test the key issues of common sense. However, these problems do not show clearly the situations that are commonly faced by systems for Ambient Intelligence and it therefore makes it more complicated to make comparisons between Ambient Intelligence systems, which in any case have not been specifically built from a common-sense point of view. In order to overcome these drawbacks, this work presents a set of benchmark problems, organized in such a way that each benchmark problem addresses a specific key issue of common-sense.

The amount of knowledge stored in the proposed approach is far from being comparable with projects such as Open Mind or Cyc and has been mainly introduced by the author of this thesis. Since it is a known fact that common-sense knowledge, far from being unique, is closely connected with personal characteristics such as age, gender, or socio-cultural factors, these aspects have to be considered when assessing the performance of the system. In order to do this, a set of 43 people have been questioned about the benchmark problems with a twofold aim. On the one hand, people have been asked to interpret the set of events corresponding to each of the described scenarios. And on the other hand, the system interpretations and responses have been evaluated by the same people.

The data analysis process has led to relevant conclusions in supporting the main working hypothesis postulated at the beginning of this thesis. One of these conclusions has shown the need to improve the reasoning mechanism when spatio-temporal and simultaneous event issues are involved. In a general sense, poor reasoning abilities were postulated as the cause for failing context interpretations. Particularly, it can be said that this working hypothesis is supported by the assessment of the respondents' satisfaction with the system interpretations and reactions, especially when it comes to spatio-temporal and simultaneous events. This is therefore interpreted as a system shortcoming, and further improvements are required in this regard.

The hypothesis about the key role played by common-sense knowledge in achieving Ambient Intelligence is supported in the test run of the prototype version. If this premise holds for a reduced but representative enough sub-set of common-sense knowledge, it can therefore be generalized to a bigger set. In this sense, it is concluded that providing Ambient Intelligence systems with common-sense knowledge is the basic requirement for those systems to be flexible enough to address the scenarios envisaged in [35].

---

[4] Some implementation details can be found at http://arco.esi.uclm.es/ mariaj.santofimia/

Finally, the causal explanation approach followed by humans demonstrates the importance that implementing an action theory has in enabling the system capability to understand and reason about what activities are being carried out in the context.

Chapter 9

Conclusions and future work

> **"A conclusion is the place where you got tired of thinking."**
>
> – *Arthur Bloch*

Summary – *This chapter summarizes the overall conclusions of this thesis, its main contributions, achievements, and outcomes. Future lines of research are also described as a continuation of the research work presented here.*

9.1 Introduction

Previous work reviewed in this thesis has brought to light the shortcomings associated with approaches for Ambient Intelligence which have overlooked the importance of leveraging common-sense abilities. In this sense, this thesis was motivated by the theory of the enabling role that common-sense knowledge has in achieving similar scenarios to those envisaged in [35].

Chapter 1 postulates specific reasons for this under-achievements regarding Ambient Intelligence scenarios, at least, in the way they were devised in [35]. These causes are therefore restated as the weaknesses that this thesis is intended to overcome. The goals sought, set out in Section 1.3, can be divided into two major groups; they have the purpose of either understanding the context or of interacting with it. In this sense, the listed goals can be classified into one or other of these two major goals.

- Understanding Goals:
    1. **U1:** To capture and model the common-sense knowledge involved in Ambient Intelligence.
    2. **U2:** To devise a context-sensitive mechanism to model and reason about the domain knowledge.
    3. **U3:** To provide an automated mechanism capable of capturing and understanding context events.
- Acting Goals:
    1. **A1:** To devise a model of agency that assumes responsibility for supervising and automating the behavior of an Ambient Intelligence system.

2. **A2:** To provide a service composition mechanism leveraging the system's self-sufficient capabilities.
3. **A3:** To leverage common-sense reasoning capabilities to deal with non-trivial key issues, such as non-determinism, cumulative effects of events, common-sense law of inertia, etc.

The purpose of this chapter is to discuss the conclusions and findings that have resulted from this thesis. However, before doing so and for the sake of clarity, the following items summarize the most relevant contributions of this work related to the achievement of the aforementioned goals.

- The knowledge involved in an Ambient Intelligence environment can be reduced to a common set of concepts and relationships that are always present, independently of the application domain (i.e. surveillance, Ambient Assisted Living, airports, etc.). These concepts and relationships basically center on the notion of *action* and *event*. In this sense, goal **U1** has been achieved by proposing a semantic model for actions and events. This model is described and formalized in Section 4.4.

- Goal **U2** basically seeks how to gather and process sensor information so that it can be translated into semantically enriched knowledge. In order to address this challenge, a Multi-Agent System approach is adopted. Additionally, the communication issues supporting the Multi-Agent System activity rely on a middleware framework. Goal **U2** is therefore addressed by means of a technological solution, however, it is also worth mentioning that both the Multi-Agent System and the middleware framework that supports it implement the semantic model. The fact that both modules implement the same semantic module articulates the mechanisms that enable the interpretation process to be undertaken on the collected data measurements.

- Goal **U3** establishes the notion of *situation* as the context in which actions and events are framed. The task of recognizing and understanding an ongoing situation consists in combining previous knowledge about the context and the information collected from the environmental sensors. Recognizing actions consists in performing matching procedures between the state of affairs before the action takes place and afterward. To this end, this thesis proposes a possible-world theory articulated by means of a multiple context mechanism as described in Chapter 5.

- Goal **A1** is connected to goal **U2** in the sense that both have been addressed by means of the same Multi-Agent System approach, although implemented in different intelligent agents. Chapter 6 describes the insights of an approach intended to generate behavioral responses complying with general principles or goals that should drive environmental behavior. To this end, a Belief, Desire, and Intention (BDI) model of agency is proposed. The most relevant aspect that distinguishes the proposed BDI model of agency from other existing models is grounded in the abstract terms in which the agents are described. Rather than providing the agents with specific behavioral patterns, they are provided with general guidelines that become concrete according to the ongoing situation and the domain knowledge.

- Goal **A2** arises as a consequence of goal **A1**, since behavioral responses not only need to be generated, they also they have to be performed in terms of the available services. The high dynamism of the devices that characterize these types of context makes it unfeasible to take for granted that a specific service will be available in the near future. It is therefore necessary to overcome this limitation by decoupling the behavioral responses from the services that will be used in their implementation. In order to do this, an automatic service composition mechanism is proposed to work around the service shortage by composing the functionality required in several cases.

- In order to address goal **A3** the role played by the Scone Knowledge-Base system has been essential. In this sense, the validation section of this thesis demonstrates how goal **A3** has been successfully addressed by elaborating a set of benchmark problems in which each of the proposed scenarios is especially adapted to testing a specific common-sense key issue.

These goals are clearly intended to set out a theoretical approach to constructing Ambient Intelligence systems capable of acting in an unsupervised manner, as envisaged in [35]. There is clear evidence that overlooking the need for common-sense knowledge is the reason why general intelligent systems, and in particular Ambient Intelligence systems, are not succeeding in their endeavor. Both the given working hypothesis and the goals pursued constitute the starting point of this thesis.

This chapter is dedicated to summarizing all the interim conclusions stated throughout the previous chapters. To this end, the following section presents the development of the research methodology used, describing all its different stages and all the results and conclusions reached. The third section describes how this thesis has contributed to the state of the art in Ambient Intelligence, listing some of the most relevant papers, published in recognized peer-reviewed forums. Finally, the last section discusses the open research lines, presented as the continuation of this work.

9.2 Summary of the results and findings

Managing Ambient Intelligence environments is undoubtedly a very challenging task, mainly due to the fact that the different scenarios that might occur in a supervised environment cannot be predicted in advance. There are countless factors that might affect the context and that cannot be foreseen or predicted. It is therefore sensible to think that the supervision of Ambient Intelligence environments has to be grounded in the ability to understand the context and to react to it according to certain goals and desires. This was therefore postulated as the key working hypothesis of this thesis.

This early hypothesis works as the starting point for this research, from which subsequent challenges arise. In this sense, the first requirement tackled has to do with how to gather information from supervised contexts. Given that these sorts of environments are populated with electronic devices, they provide appropriate means to capture the events that are taking place in them. A middleware solution, that extends the work in [144] and [146] and, with enhanced capabilities to support the context modeling and reasoning tasks, is proposed here to meet the stated requirements successfully. However, despite the fact that the middleware solution is in charge of all the communication issues, additional mechanisms are also required to automate the filtering and processing of the collected information.

In order to tackle this new need, the use of a Multi-Agent System is proposed to take responsibility for supervising and processing the events that take place in the supervised environment. For example, a presence sensor activation might be meaningless if the sensor is located in a busy corridor. However, if that sensor is located in a restricted area, it might be suggestive of a subversive action. The Multi-Agent System, appropriately configured to do so, is in charge of discerning between events that need action treatment and those that are irrelevant.

The most appropriate model of agency, among all those that have been proposed to date, is that of Belief, Desire, and Intention (BDI). This model seems to be the most suitable, since the Multi-Agent

System has to be motivated by goals and desires, and in order to do so, agents need plans to achieve their aims.

It soon becomes apparent that a solution based on a Multi-Agent System will not suffice to deal with unforeseen situations. Novel situations need to be tackled by enabling the system capability to devise *ad-hoc* plans that adapt the current context situation to the targeted one. In this regard, the pre-coded plans proposed by the BDI model of agency are not sufficient to satisfy the requirements.

A rough analysis of what peculiarity differentiates people from animals brings to light that only by considering common sense as a constituent part of the system can it be enabled to address novel situations autonomously. At this stage of the research, the common sense theory becomes one of the cornerstones of the thesis.

Research efforts are therefore directed at endowing agents with common-sense reasoning capabilities that enable them to devise plans in an *ad-hoc* manner. Among the few solutions that, to date, can be found with systems providing common-sense knowledge and reasoning capabilities, Scone appears as the most suitable solution. In this sense, the responsibility to figure out the most appropriate plan for a certain agent goal was given to Scone, which on the basis of an abstract action, to be performed upon a certain object, was capable of planning the set of services that, once composed, could behave as a composite service to drive the context to the goal state.

The introduction of common-sense knowledge brings a new challenge, the need for a context model that provides a common nomenclature between the Multi-Agent System, in charge of capturing the environmental events, and the knowledge-base system, responsible for holding the representation of such events. In this sense, independently of the considered contexts (a home environment, an office, a train station, etc.) the only information that is relevant for the system supervising the environment is that related to the events that take place in it. This peculiarity is what motivates the need to formalize the semantic model for actions and events.

The research methodology followed is finally evaluated by building a prototype implementation of the system, as described in Chapter 8. Aside from the prototype insights, that chapter also offers an evaluation methodology by which the services provided by the system can be assessed in terms of the degree of user satisfaction. A sample scenario is represented by means of synthesized events and services, in which three different cases are considered: a) only basic services are provided; b) basic and composite services are provided; c) only composite services are provided. Using the evaluation methodology proposed in **??**, and for the paradigmatic case scenario considered for the evaluation, it can be concluded that composite services tend to comply most closely with user preferences and needs, compared to services that simply involve basic or a combination of both types of services. This conclusion can be generalized into a broader statement affirming that tailored service solutions more adequately comply with the flexibility required to address novel situations.

Beside the prototype implementation, the working hypothesis and the initial research questions have also been evaluated by a pair of questionnaires. The first questionnaire is aimed at comparing the system's interpretations and responses with those provided by humans, given a certain set of events. The second questionnaire assesses how satisfied respondents are with the system's performance in each of the nine case scenarios presented.

Considering all the conclusions drawn from the construction of the prototype and the analysis of the questionnaire results, it can be finally concluded that the capacity of Ambient Intelligence systems to address novel situations is a direct consequence of the amount of common-sense knowledge provided to the system. The more knowledge it holds, the more capable it is of flexibly adapting its behavior to comply with the desired goals. At the same time, this is one of the main drawbacks of this approach, since system performance is directly dependent on how much knowledge is held about a certain knowledge domain involved in a specific event occurrence.

Furthermore, an additional shortcoming of this work is that all of the summarized conclusions are the consequence of a generalization process. In this sense, could be claimed that there is no real evidence supporting these conclusions, other than that those conclusions are drawn from a paradigmatic sample. It can therefore be argued that a thorough evaluation of all the different scenarios is unfeasible, given the countless number of factors that might affect a given scenario. For that reason, the adopted approach consists of taking a representative number of different scenarios to run the evaluation tests upon them, in such a way that the conclusions achieved from the analysis of these scenarios can be generalized to a broader spectrum of untested scenarios.

9.3 Publications

The different stages through which this thesis has evolved have given rise to several publications, some of which have already been mentioned. The complete list is presented bellow:

- The article "*Possible-World and Multiple-Context Semantics for Common-Sense Action Planning*", published in the Space, Time and Ambient Intelligence Workshop held at the 2011 International Joint Conference on Artificial Intelligence (IJCAI) [125], is intended to propose an approach for action planning with endowed capabilities to handle the non-trivial aspects of common-sense reasoning (goal **U3**).

- The paper entitled "*A Semantic Model for Actions and Events in Ambient Intelligence* " [123], accepted for publication in the Journal of Engineering Applications of Artificial Intelligence, proposes handling environmental actions and events through the use of a semantic model for Ambient Intelligence which, under the umbrella of a philosophical and common sense perspective, describes what actions and events are considered, how they are connected, and how computational systems should think about their meaning (goal **U1**).

- The article "*A Common-Sense Planning Strategy for Ambient Intelligence*" [124] has been presented in the 14th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems. By combining and supporting an action planner on a common sense system, this paper proposes an approach that overcomes some of the deficiencies of the current approaches for Ambient Intelligence systems (goal **A1**).

- The paper entitled "*A distributed architectural strategy towards ambient intelligence*" [126], published in the IWANN '09 Proceedings of the 10th International Work-Conference on Artificial Neural Networks, was mainly devoted to presenting the potential benefits obtained from combining common-sense reasoning and multi-agent systems with a fully equipped middleware platform (goal **U2**).

- The paper "*An Agent-based approach towards Automatic Service Composition in Ambient Intelligence*" [127], presented in the Intelligent Agent Technology (IAT) Conference, proposes a first approach to the research work presented here, grounded in the use of artificial intelligent agents for the automation of the composition task (goal **A2**).

9.4 Future work

Achieving automatic service composition, as a means to leverage the system's capability to autonomously supervise Ambient Intelligence environments, is a task that can be considered to have been addressed by this thesis. However, the results obtained lead to the need for further work to be conducted as a continuation of this thesis.

Despite the fact that the work in [144] and [146] made important contributions to the goal of automatically incorporating different technologies into the communication platform, there still exist technologies that have not so far been incorporated into the communication architecture. This limitation poses the important challenge of providing Ambient Intelligence systems with the capability to make use of all the different devices present in the context.

The proposed architecture has resorted to a BDI model of agency for implementing the cognitive capacities of the system. However, those agents have been externally enhanced with common-sense reasoning capabilities and knowledge. In this sense, it would be desirable to count on common-sense agents that do not have to look for reasoning capabilities and knowledge externally.

Additionally, due to the fact that Scone does not count on an extensive knowledge base, as Cyc and OpenCyc do, it has been necessary to introduce all the possible actions one by one, in terms of their multiple contexts, before, after, and during the action. In this sense, it would be desirable to automate this task, adopting one or other of the strategies that, at CycCorp, are being tested to retrieve from the Internet all the common-sense knowledge that a normal person holds.

Rather than facing such an ambitious project as in CycCorp, the idea would be basically intended to retrieve information about actions and events in an autonomous way. The limited set of actions considered by the prototype implementation is the main limitation of the current approach. In this sense, if a dictionary of actions could be introduced into the knowledge-base, the ability of the system to understand and to devise behavioral responses would be improved.

Finally, the ultimate goal is to achieve a system capable of knowing how to behave under different domains, without being explicitly told. For example, for an Ambient Intelligence system devoted to supervising a train station, one of the goals of the system is trying to determine whether a piece of luggage has been left unattended. However, this goal has been given to the system, as well as the general guidelines that dictate how the system should proceed. In this sense, the opened future line poses the challenge of elaborating responses without explicitly requiring these abstract guidelines to be provided.

Part V

References

Bibliography

[1] Gregory D. Abowd, Anind K. Dey, Peter J. Brown, Nigel Davies, Mark Smith, and Pete Steggles. Towards a better understanding of context and context-awareness. In *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, HUC '99, pages 304–307, London, UK, 1999. Springer-Verlag.

[2] Varol Akman. Book review: Formalizing Common Sense: Papers by John McCarthy edited by Vladimir Lifschitz (Ablex, 1990). *SIGART Bull.*, 4:12–13, January 1993.

[3] Varol Akman and Mehmet Surav. The use of situation theory in context modeling. *Computational Intelligence*, 13(3):427–438, 1997.

[4] James F. Allen. Towards a general theory of action and time. *Artif. Intell.*, 23:123–154, July 1984.

[5] Oliver Amft, Clemens Lombriser, Thomas Stiefmeier, and Gerhard Tröster. Recognition of user activity sequences using distributed event detection. In *Proceedings of the 2nd European conference on Smart sensing and context*, EuroSSC'07, pages 126–141, Berlin, Heidelberg, 2007. Springer-Verlag.

[6] Francesco Amigoni, Nicola Gatti, Carlo Pinciroli, and Manuel Roveri. What planner for Ambient Intelligence applications? *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 35(1):7–21, 2005.

[7] Alexander Artikis, Marek Sergot, and Georgios Paliouras. A logic programming approach to activity recognition. In *Proceedings of the 2nd ACM international workshop on Events in multimedia*, EiMM '10, pages 3–8, New York, NY, USA, 2010. ACM.

[8] Juan Carlos Augusto, Hideyuki Nakashima, and Hamid Aghajan. Ambient Intelligence and Smart Environments: A State of the Art. In *Handbook of Ambient Intelligence and Smart Environments*, pages 3–31. 2010.

[9] Ayomi Bandara, Terry Payne, David De Roure, Nicholas Gibbins, and Tim Lewis. A pragmatic approach for the semantic description and matching of pervasive resources. In *GPC'08: Proceedings of the 3rd international conference on Advances in grid and pervasive computing*, pages 434–446, Berlin, Heidelberg, 2008. Springer-Verlag.

[10] Jon Barwise and John Perry. Situations and attitudes. *Journal of Philosophy*, 78(11):668–691, 1981.

[11] Claudio Bettini, Oliver Brdiczka, Karen Henricksen, Jadwiga Indulska, Daniela Nicklas, Anand Ranganathan, and Daniele Riboni. A survey of context modelling and reasoning techniques. *Pervasive Mob. Comput.*, 6:161–180, April 2010.

[12] Ronald Brachman and Hector Levesque. *Knowledge Representation and Reasoning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[13] Michael Bratman. *Intentions, Plans, and Practical Reason*. Harvard University Press, 1987.

[14] Phillip Bricker. David Lewis: On the Plurality of Worlds. *Central Works of Philosophy. The Twentieth Century: Quine and After*, 2006.

[15] J. Bronstead, K. M. Hansen, and M. Ingstrup. A Survey of Service Composition Mechanisms in Ubiquitous Computing. In *Second Workshop on Requirements and Solutions for Pervasive Software Infrastructures, UbiComp 2007 Workshops Proceedings*, 2007.

[16] Roy H. Campbell. A middleware for context-aware agents in ubiquitous computing environments. In *in ACM/IFIP/USENIX International Middleware Conference, Rio de*, pages 16–20, 2003.

[17] Jian Cao, Jie Wang, Shensheng Zhang, and Minglu Li. A dynamically reconfigurable system based on workflow and service agents. *Eng. Appl. Artif. Intell.*, 17(7):771–782, 2004.

[18] H. Chen. *An intelligent broker architecture for pervasive context-aware systems*. PhD thesis, University of Maryland, Baltimore County, 2004.

[19] Harry Chen, Tim Finin, and Anupam Joshi. An Ontology for Context-Aware Pervasive Computing Environments. *Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review*, 18:197–207, 2003.

[20] Harry Chen, Tim Finin, and Anupam Joshi. Semantic Web in the Context Broker Architecture. In *Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications (PerCom'04)*, PERCOM '04, pages 277–, Washington, DC, USA, 2004. IEEE Computer Society.

[21] Harry Chen, Filip Perich, Tim Finin, and Anupam Joshi. Soupa: Standard ontology for ubiquitous and pervasive applications. In *In International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pages 258–267, 2004.

[22] Wei Chen and Scott E. Fahlman. Modeling mental contexts and their interactions. In *AAAI 2008 Fall Symposium on Biologically Inspired Cognitive Architectures, Washington*. 2008.

[23] Aurelie Clodic, Rachid Alami, Vincent Montreuli, Shuyin Li, Britta Wrede, and Agnes Swadzba. A study of interaction between dialog and decision for human-robot collaborative task achievement, 26/08/2007 2007.

[24] Cogniron. COGNIRON Project RA3 Final Deliverable. Technical report, LAAS/CNRS, April 2008.

[25] Diane J. Cook, Juan C. Augusto, and Vikramaditya R. Jakkula. Ambient Intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, 5(4):277–298, August 2009.

[26] P. Costa, G. Coulson, C. Mascolo, L. Mottola, G. P. Picco, and S. Zachariadis. A reconfigurable component-based middleware for networked embedded systems. *Int. Journal of Wireless Information Networks*, 14(2), 2007.

[27] Gianpaolo Cugola and Gian Pietro Picco. Reds: a reconfigurable dispatching system. In *SEM '06: Proceedings of the 6th international workshop on Software engineering and middleware*, pages 9–16, New York, NY, USA, 2006. ACM.

[28] Cycorp. The ontological engineer's handbook. version 0.7, 2002.

[29] Donald Davidson. The Individuation of Events. *Essays on Actions and Events*, 1(9):163–181.

[30] Donald Davidson. Actions, reasons, and causes. *The Journal of Philosophy*, 60(23):685–700, 1963.

[31] Oleg Davidyuk, Nikolaos Georgantas, Valérie Issarny, and Jukka Riekki. MEDUSA: Middleware for End-User Composition of Ubiquitous Applications. In F. Mastrogiovanni and N.Y. Chong, editors, *Handbook of Research on Ambient Intelligence and Smart Environments: Trends and Perspectives*. IGI Global, 2010.

[32] Renato de Freitas Bulcao Neto and Maria da Graca Campos Pimentel. Toward a domain-independent semantic model for context-aware computing. *Web Congress, Latin American*, 0:61–70, 2005.

[33] Lucia Del Prete and Licia Capra. Mosca: seamless execution of mobile composite services. In *Proceedings of the 7th workshop on Reflective and adaptive middleware*, ARM '08, pages 5–10, New York, NY, USA, 2008. ACM.

[34] John Divers. *Possible worlds. Problems of philosophy*. Routledge, 2002.

[35] K. Ducatel, M. Bogdanowicz, F. Scapolo, J. Leijten, and J. C. Burgelman. ISTAG: Scenarios for Ambient Intelligence in 2010. Technical report, ISTAG, 2001.

[36] Markus Eisenhauer, Peter Rosengren, and Pablo Antolin. HYDRA: A Development Platform for Integrating Wireless Devices and Sensors into Ambient Intelligence Systems. In *The Internet of Things: 20th Tyrrhenian Workshop on Digital Communications*, pages 367–373. 2010.

[37] H. R. Ekbia. *Artificial Dreams: The Quest for Non-Biological Intelligence*. Cambridge University Press, New York, NY, USA, 1st edition, 2008.

[38] Kutluhan Erol, James Hendler, and Dana S. Nau. HTN planning: Complexity and expressivity. In *In AAAI-94*, 1994.

[39] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

[40] Scoott E. Fahlman. Marker-passing inference in the scone knowledge-base system. In *First International Conference on Knowledge Science, Engineering and Management (KSEM'06)*. Springer-Verlag (Lecture Notes in AI), 2006.

[41] Scott E. Fahlman. *NETL: A System for Representing and Using Real-World Knowledge*. MIT Press, Cambridge, MA, 1979.

[42] Scott E. Fahlman. The Scone knowledge-base project, 2010. Available online at: `http://www.cs.cmu.edu/ sef/scone/`. Retrieved on February 28th, 2010.

[43] Christiane Fellbaum. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication).* The MIT Press, May 1998.

[44] Álvaro Fides-Valero, Matteo Freddi, Francesco Furfari, and Mohammad-Reza Tazari. The persona framework for supporting context-awareness in open distributed systems. In *Proceedings of the European Conference on Ambient Intelligence*, AmI '08, pages 91–108, Berlin, Heidelberg, 2008. Springer-Verlag.

[45] Christopher Foley, Gemma Power, Leigh Griffin, Chen Chen, Niall Donnelly, and Eamonn de Leastar. Service Group Management facilitated by DSL driven Policies in embedded Middleware. In *IEEE ISCC Conference*, June 2010.

[46] Foundation for Intelligent Physical Agents (FIPA). FIPA ACL Message Structure Specification, 2002. Available online at: `http://www.fipa.org/specs/fipa00061/SC00061G.html`. Retrieved on February 28th, 2010.

[47] A.U. Frank. Tiers of ontology and consistency constraints in Geographic Information Systems. *International Journal of Geographical Information Science (IJGIS)*, 5:667–678, 2001.

[48] Keita Fujii and Tatsuya Suda. Component service model with semantics (cosmos): A new component model for dynamic service composition. In *Proceedings of the 2004 Symposium on Applications and the Internet-Workshops (SAINT 2004 Workshops)*, SAINT-W '04, pages 348–, Washington, DC, USA, 2004. IEEE Computer Society.

[49] Nicola Gatti, Francesco Amigoni, and Marco Rolando. Multiagent Technology Solutions for Planning in Ambient Intelligence. In *WI-IAT '08: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 286–289, Washington, DC, USA, 2008. IEEE Computer Society.

[50] Michael P. Georgeff. A theory of action for multiagent planning. In A. H. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 205–209. Kaufmann, San Mateo, CA, 1988.

[51] Anders Green. Characterising dimensions of use for designing adaptive dialogues for human-robot communication. In *RO-MAN*, pages 1078–1083, 2007.

[52] Pavel Gregoric. *Aristotle on the Common Sense.* Oxford University Press, 2007.

[53] Matthias Grossmann, Martin Bauer, Nicola Honle, Uwe-Philipp Kappeler, Daniela Nicklas, and Thomas Schwarz. Efficiently managing context information for large-scale scenarios. In *Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications*, pages 331–340, Washington, DC, USA, 2005. IEEE Computer Society.

[54] Tao Gu, Hung Keng Pung, and Da Qing Zhang. A middleware for building context-aware mobile services. In *In Proceedings of IEEE Vehicular Technology Conference (VTC*, 2004.

[55] Tao Gu, Hung Keng Pung, and Da Qing Zhang. A service-oriented middleware for building context-aware services. *J. Netw. Comput. Appl.*, 28:1–18, January 2005.

[56] Ramanathan Guha. *Contexts: a formalization and some applications*. PhD thesis, Stanford, CA, USA, 1992. UMI Order No. GAX92-17827.

[57] Patrick J. Hayes. The second naive physics manifesto. In *Computation & intelligence*, pages 567–585. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1995.

[58] Karen Henricksen and Jadwiga Indulska. Modelling and using imperfect context information. In *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, PERCOMW 04, pages 33–, Washington, DC, USA, 2004. IEEE Computer Society.

[59] Karen Henricksen and Jadwiga Indulska. Developing context-aware pervasive computing applications: Models and approach. *Pervasive Mobile Computing*, 2:37–64, February 2006.

[60] Karen Henricksen, Jadwiga Indulska, and Andry Rakotonirainy. Modeling context information in pervasive computing systems. In *Proceedings of the First International Conference on Pervasive Computing*, Pervasive 02, pages 167–180, London, UK, 2002. Springer-Verlag.

[61] A. Herbon, A. Oehme, and E. Zentsch. Emotions in Ambient Intelligence - An experiment on how to measure affective states. In *20th BCS HCI Group Conference*. University of London, 2006.

[62] B. Hommel, Jochen Musseler, Gisa Aschersleben, and Wolfgang Prinz. The theory of event coding (TEC): A framework for perception and action planning. *Behavioral and Brain Sciences*, 24:849–878, 2001.

[63] Laurence R. Horn and Gregory Ward. *The Handbook of Pragmatics (Blackwell Handbooks in Linguistics)*. Blackwell Publishers, 2004.

[64] Jan Hreno, Peter Kostelnik, Martin Sarnovsky, Matts Ahlsen, Peter Rosengren, Peeter Kool, and Mathias Axling. Semantic Devices for Ambient Environment Middleware. In *The Internet of Things and Services Research Workshop*, European Research Workshops Proceedings, 2008.

[65] Valérie Issarny, Daniele Sacchetti, Ferda Tartanoglu, Françoise Sailhan, Rafik Chibout, Nicole Levy, and Angel Talamona. Developing Ambient Intelligence Systems: A Solution based on Web Services. *Automated Software Engg.*, 12:101–137, January 2005.

[66] K.G. Jolly, R. Sreerama Kumar, and R. Vijayakumar. Intelligent task planning and action selection of a mobile robot in a multi-agent system through a fuzzy neural network approach. *Engineering Applications of Artificial Intelligence*, 23(6):923–933, 2010.

[67] Glenn Judd and Peter Steenkiste. Providing contextual information to pervasive computing applications. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, PERCOM03, pages 133–, Washington, DC, USA, 2003. IEEE Computer Society.

[68] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1995.

[69] Dongyeop Kang and Ho-Jin Choi. Robot task planning for mixed-initiative human robot interaction in home service robot. In *CITWORKSHOPS '08: Proceedings of the 2008 IEEE 8th International Conference on Computer and Information Technology Workshops*, pages 49–54, Washington, DC, USA, 2008. IEEE Computer Society.

[70] Julia Kantorovitch, Jarmo Kalaoja, Ilkka Niskanen, and Toni Piirainen. Towards a Better Understanding of Semantic Ontology-Based Home Service Modelling. *Advanced Information Networking and Applications, International Conference on*, 0:26–31, 2008.

[71] Eunhoe Kim and Jaeyoung Choi. A semantic interoperable context infrastructure using web services. In *ICCSA'07: Proceedings of the 2007 international conference on Computational science and Its applications*, pages 839–848, Berlin, Heidelberg, 2007. Springer-Verlag.

[72] Donald E. Knuth. Semantics of context-free languages. *Theory of Computing Systems*, 2(2):127–145, June 1968.

[73] R Kowalski and M Sergot. A logic-based calculus of events. *New Generation Comput.*, 4:67–95, January 1986.

[74] Saul Aaron Kripke. A semantical analysis of modal logic I: Normal modal propositional calculi. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 9:67–96, 1963.

[75] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Also available at http://planning.cs.uiuc.edu/.

[76] Doug Lenat, Mayank Prakash, and Mary Shepherd. CYC: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks. *Artificial Intelligence Magazine*, 6:65–85, January 1986.

[77] Douglas Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38:33–38, 1995.

[78] Douglas B. Lenat and Edward A. Feigenbaum. On the thresholds of knowledge. In *IJCAI*, pages 1173–1182, 1987.

[79] Douglas B. Lenat and R. V. Guha. *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[80] Douglas B. Lenat, R. V. Guha, Karen Pittman, Dexter Pratt, and Mary Shepherd. CYC: toward programs with common sense. *Commun. ACM*, 33(8):30–49, 1990.

[81] Rhys Lewis and José Manuel Cantera Fonseca. Delivery context ontology. World Wide Web Consortium, Working Draft WD-dcontology-20080415, April 2008.

[82] Hugo Liu and Push Singh. Conceptnet: A practical commonsense reasoning toolkit. *BT Technology Journal*, 22:211–226, 2004.

[83] Clemens Lombriser, Nagendra B. Bharatula, Daniel Roggen, and Gerhard Tröster. On-body activity recognition in a dynamic sensor network. In *BodyNets '07: Proceedings of the ICST 2nd international conference on Body area networks*, pages 1–6, ICST, Brussels, Belgium,

Belgium, 2007. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[84] Edith Maier and Guido Kempter. AAL in the wild — lessons learned. In *Proceedings of the 5th International on ConferenceUniversal Access in Human-Computer Interaction. Part II: Intelligent and Ubiquitous Interaction Environments*, UAHCI '09, pages 218–227, Berlin, Heidelberg, 2009. Springer-Verlag.

[85] Edith Maier and Guido Kempter. ALADIN - a Magic Lamp for the Elderly? In Hideyuki Nakashima, Hamid Aghajan, and Juan Carlos Augusto, editors, *Handbook of Ambient Intelligence and Smart Environments*, pages 1201–1227. Springer US, 2010.

[86] David L. Martin, Massimo Paolucci, Sheila A. McIlraith, Mark H. Burstein, Drew V. McDermott, Deborah L. McGuinness, Bijan Parsia, Terry R. Payne, Marta Sabou, Monika Solanki, Naveen Srinivasan, and Katia P. Sycara. Bringing Semantics to Web Services: The OWL-S Approach. In *SWSWPC*, pages 26–42, 2004.

[87] J. McCarthy. Ascribing mental qualities to machines. In M. Ringle, editor, *Philosophical Perspectives in Artificial Intelligence*, pages 161–195. Humantities Press, 1979.

[88] J. McCarthy and P. J. Hayes. *Some philosophical problems from the standpoint of Artificial Intelligence*, pages 26–45. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

[89] John McCarthy. Programs with common sense. In *Semantic Information Processing*, volume 1, pages 403–418, 1968.

[90] John McCarthy. Notes on formalizing context. In *Proceedings of the 13th international joint conference on Artifical intelligence - Volume 1*, pages 555–560, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[91] John Mccarthy. A Logical AI Approach to Context. unpublished. 1996.

[92] Rene Meier and Vinny Cahill. STEAM: Event-based middleware for wireless ad hoc networks. In *Proceedings of the 1st International Workshop on Distributed Event-Based Systems (DEBS02), Vienna, Austria*, 2002.

[93] Marius Mikalsen, Sten Hanke, Thomas Fuxreiter, Ståle Walderhaug, and Leendert W. M. Wienhofen. Interoperability services in the MPOWER Ambient Assisted Living Platform. In *MIE*, pages 366–370, 2009.

[94] George A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38:39–41, 1995.

[95] Marvin Minsky. *The society of mind*. Simon & Schuster, Inc., New York, NY, USA, 1986.

[96] Marvin Minsky. Society of mind: A response to four reviews. *Artif. Intell.*, 48(3):371–396, 1991.

[97] Marvin Minsky. The emotion machine: from pain to suffering. In *Creativity & Cognition*, pages 7–13, 1999.

[98] R. C. Moore. A Formal Theory of Knowledge and Action. In J. R. Hobbs and R. C. Moore, editors, *Formal Theories of the Commonsense World*, pages 319–358. Ablex, Norwood, NJ, 1985.

[99] Martin M. Morandell, Andreas Hochgatterer, Sascha Fagel, and Siegfried Wassertheurer. Avatars in assistive homes for the elderly. In *Proceedings of the 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society on HCI and Usability for Education and Work*, USAB '08, pages 391–402, Berlin, Heidelberg, 2008. Springer-Verlag.

[100] Erik T. Mueller. *Commonsense Reasoning*. Morgan Kaufmann, 2006.

[101] Xenia Naidenova and Xenia Naidenova. *Machine Learning Methods for Commonsense Reasoning Processes: Interactive Models*. Information Science Reference - Imprint of: IGI Publishing, Hershey, PA, 2009.

[102] D. Nicklas and B. Mitschang. The nexus augmented world model: An extensible approach for mobile, spatially-aware applications. In *Proceedings of the 7th International Conference on Object-Oriented Information Systems*, pages 392–401, 2001.

[103] William T. Niu and Judy Kay. PERSONAF: framework for personalised ontological reasoning in pervasive computing. *User Modeling and User-Adapted Interaction*, 20(1):1–40, 2010.

[104] Kathy Panton, Cynthia Matuszek, Douglas Lenat, Dave Schneider, Michael Witbrock, Nick Siegel, and Blake Shepard. Common sense reasoning - from Cyc to intelligent assistant. In Yang Cai and Julio Abascal, editors, *Ambient Intelligence in Everyday Life, LNAI 3864*, pages 1–31. Springer, 2006.

[105] A. Parmar. The representation of actions in KM and CYC (tech. rep. no. FRG-1). Technical report, Standford, CA: Department of Computer Science, Standford University, 2001.

[106] Theodore Patkos, Antonis Bikakis, Grigoris Antoniou, Maria Papadopouli, and Dimitris Plexousakis. Distributed AI for Ambient Intelligence: Issues and Approaches. In Bernt Schiele, Anind K. Dey, Hans Gellersen, Boris E. R. de Ruyter, Manfred Tscheligi, Reiner Wichert, Emile H. L. Aarts, and Alejandro P. Buchmann, editors, *AmI*, volume 4794 of *Lecture Notes in Computer Science*, pages 159–176. Springer, 2007.

[107] A. Pitsillides, E. Themistokleous, G. Samaras, S. Walderhaug, and O. M. Winnem. Overview of MPOWER: Middleware platform for the cognitively impaired and elderly. In *IST-Africa 2007 Conference*, May 2007.

[108] Maria Poveda-Villalon and Raul Garcia-Castro. Entregable 1.1.3 Formalización e implementación de las ontologías mIO! Technical report, Ontology Engineering Group. Universidad Politecnica de Madrid, 2009.

[109] Mirko Presser, Alexander Gluhak, Derek Babb, Rahim Tafazolli, and Laurent Hérault. e-SENSE - capturing Ambient Intelligence for Mobile Communications through wireless sensor networks. In *WINSYS*, pages 341–346, 2006.

[110] Davy Preuveneers, Jan Van den Bergh, Dennis Wagelaar, Andy Georges, Peter Rigole, Tim Clerckx, Yolande Berbers, Karin Coninx, Viviane Jonckers, and Koen De Bosschere. *Towards an Extensible Context Ontology for Ambient Intelligence*. 2004.

[111] Giuseppe Primiero. *Information and Knowledge. A Constructive Type-Theoretical Approach. Logic Epistemology and the Unity of Science*. Springer, Berlin.

[112] Giuseppe Primiero. Information and knowledge. a constructive type-theoretical approach. In *Logic, Epistemology, and the Unity of Science*, volume 10. Springer, Berlin, 2008.

[113] Carlos Ramos, J. C. Augusto, and Daniel Shapiro. Ambient Intelligence—the Next Step for Arlosrtificial Intelligence. *IEEE Intelligent Systems*, 23(2):15–18, 2008.

[114] F. Ramparany, J. Euzenat, T.H.F. Broens, A. Bottaro, and R. Poortinga. Context management and semantic modelling for Ambient Intelligence, April 2006.

[115] Anand Ranganathan, Robert E McGrath, Roy H. Campbell, and M. Dennis Mickunas. Use of ontologies in a pervasive computing environment. *Knowl. Eng. Rev.*, 18:209–220, September 2003.

[116] Roland Reichle, Michael Wagner, Mohammad Ullah Khan, Kurt Geihs, Jorge Lorenzo, Massimo Valla, Cristina Fra, Nearchos Paspallis, and George A. Papadopoulos. A comprehensive context modeling framework for pervasive computing systems. In *DAIS'08: Proceedings of the 8th IFIP WG 6.1 international conference on Distributed applications and interoperable systems*, pages 281–295, Berlin, Heidelberg, 2008. Springer-Verlag.

[117] Carson J. Reynolds and Christopher R. Wren. Worse is better for Ambient Sensing. In *Workshop on Privacy, Trust and Identity Issues for Ambient Intelligence*, Dublin, Ireland, May 2006. Pervasive.

[118] Michael M. Richter. The search for knowledge, contexts, and case-based reasoning. *Engineering Applications of Artificial Intelligence*, 22(1):3 – 9, 2009.

[119] Manuel Roman, Christopher Hess, Renato Cerqueira, Roy H. Campbell, and Klara Nahrstedt. Gaia: A middleware infrastructure to enable active spaces. *IEEE Pervasive Computing*, 1:74–83, 2002.

[120] Manuel Roman and Nayeem Islam. Dynamically programmable and reconfigurable middleware services. In *Middleware '04: Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware*, pages 372–396, New York, NY, USA, 2004. Springer-Verlag New York, Inc.

[121] Ioanna Roussaki, Ioannis Papaioannou, Dimitrios Tsesmetzis, Julia Kantorovitch, Jarmo Kalaoja, and Remco Poortinga. Ontology Based Service Modelling for Composability in Smart Home Environments. In Max Muhlhauser, Alois Ferscha, and Erwin Aitenbichler, editors, *Constructing Ambient Intelligence*, volume 11 of *Communications in Computer and Information Science*, pages 411–420. Springer Berlin Heidelberg, 2008.

[122] N. S. Ryan, J. Pascoe, and D. R. Morse. Enhanced Reality Fieldwork: the Context-aware Archaeological Assistant. In V. Gaffney, M. van Leusen, and S. Exxon, editors, *Computer*

Applications in Archaeology 1997, British Archaeological Reports, Oxford, October 1998. Tempus Reparatum.

[123] Maria J. Santofimia, Scott E. Fahlman, Xavier del Toro, Francisco Moya, and Juan C. Lopez. A semantic model for actions and events in Ambient Intelligence. *Engineering Applications of Artificial Intelligence*, In Press, Corrected Proof:–, 2011.

[124] María J. Santofimia, Scott E. Fahlman, Francisco Moya, and Juan C. López. A common-sense planning strategy for Ambient Intelligence. In *Proceedings of the 14th international conference on Knowledge-based and intelligent information and engineering systems: Part II*, KES'10, pages 193–202, Berlin, Heidelberg, 2010. Springer-Verlag.

[125] Maria J. Santofimia, Scott E. Fahlman, Francisco Moya, and Juan C. Lopez. Possible-world and multiple-context semantics for common-sense action planning. In Mehul Bhatt, Hans W. Guesgen, and Juan Carlos Augusto, editors, *Proceedings of the Workshop on Space, Time and Ambient Intelligence, held at IJCAI-11*, IJCAI 2011 Workshop Proceedings Collection, 2011.

[126] Maria J. Santofimia, Francisco Moya, Felix Villanueva, David Villa, and Juan Lopez. A Distributed Architectural Strategy towards Ambient Intelligence. In Sigeru Omatu, Miguel Rocha, José Bravo, Florentino Fernández, Emilio Corchado, Andrés Bustillo, and Juan Corchado, editors, *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*, volume 5518 of *Lecture Notes in Computer Science*, pages 130–133. Springer Berlin / Heidelberg, 2009.

[127] Maria J. Santofimia, Francisco Moya, Felix J. Villanueva, David Villa, and Juan C. Lopez. An agent-based approach towards automatic service composition in Ambient Intelligence. *Artificial Intelligence Review*, 29:265–276, June 2008.

[128] B. N. Schilit and M. M. Theimer. Disseminating active map information to mobile hosts. *IEEE Network*, 8(5):22–32, Sept 1994.

[129] Bill Schilit, Norman Adams, and Roy Want. Context-aware computing applications. In *In Proceedings of the Workshop on Mobile Computing Systems and Applications*, pages 85–90, 1994.

[130] C. F. Schmidt. Partial provisional planning: Some aspects of commonsense planning. In J. R. Hobbs and R. C. Moore, editors, *Formal Theories of the Commonsense World*, pages 227–250. Ablex, Norwood, NJ, 1985.

[131] Purvesh Shah, David Schneider, Cynthia Matuszek, Robert C. Kahlert, Bjorn Aldag, David Baxter, John Cabral, Michael Witbrock, and Jon Curtis. Automated Population of CYC: Extracting Information about Named-entities from the Web. In *In Proceedings of the Nineteenth International FLAIRS Conference*, pages 153–158. AAAI Press, 2006.

[132] Murray Shanahan. Representing continuous change in the event calculus. In *ECAI*, pages 598–603, 1990.

[133] Push Singh, Thomas Lin, Erik T. Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. Open mind common sense: Knowledge acquisition from the general public. In *On the Move to Meaningful Internet Systems, 2002 - DOA/CoopIS/ODBASE 2002 Confederated International*

Conferences DOA, CoopIS and ODBASE 2002, pages 1223–1237, London, UK, UK, 2002. Springer-Verlag.

[134] J. F. Sowa. Crystallizing theories out of knowledge soup. In Z. W. Ras and M. Zemankova, editors, *Intelligent Systems: State of the Art and Future Directions*, pages 456–487. Ellis Horwood, New York, 1990.

[135] John F Sowa. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, Reading, MA., 1984.

[136] John F. Sowa. Syntax, semantics, and pragmatics of contexts. In *Proceedings of the Third International Conference on Conceptual Structures: Applications, Implementation and Theory*, pages 1–15, London, UK, 1995. Springer-Verlag.

[137] John F. Sowa. Peircean foundations for a theory of context. In *Proceedings of the Fifth International Conference on Conceptual Structures: Fulfilling Peirce's Dream*, ICCS '97, pages 41–64, London, UK, 1997. Springer-Verlag.

[138] Robert Speer. Open Mind Commons: An Inquisitive Approach to Learning Common Sense. In *Workshop on Common Sense and Intelligent User Interfaces*, 2007.

[139] Thomas Strang and Claudia L. Popien. A context modeling survey. In *UbiComp 1st International Workshop on Advanced Context Modelling, Reasoning and Management*, pages 31–41, Nottingham, September 2004.

[140] Ashley W. Stroupe and Tucker R. Balch. Value-based action selection for observation with robot teams using probabilistic techniques. *Robotics and Autonomous Systems*, 50(2-3):85–97, 2005.

[141] Mohammad-Reza Tazari, Francesco Furfari, Juan-Pablo Lázaro Ramos, and Erina Ferro. The PERSONA service platform for AAL spaces. In *Handbook of Ambient Intelligence and Smart Environments*, pages 1171–1199. 2010.

[142] Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter, editors. *Handbook of Knowledge Representation (Foundations of Artificial Intelligence)*. Elsevier Science, 2007.

[143] P. Verissimo, V. Cahill, A. Casimiro, K. Cheverst, A. Friday, and J. Kaiser. CORTEX: Towards supporting autonomous and cooperating sentient entities. In *Proceedings of European Wireless 2002*, pages 595–601, Florence, Italy, 2002.

[144] David Villa. *Infraestructura para la integracion de redes de sensores y actuadores en entornos inteligentes*. PhD thesis, University of Castilla-La Mancha, Spain, December 2009.

[145] Maria Poveda Villalon, MC. Suárez-Figueroa, and A. Gómez-Pérez. Reusing ontology design patterns in a context ontology network. In *Proceedings Second Workshop on Ontology Patterns (WOP 2010) co-located at ISWC 2010*, volume 671, Alemania, Noviembre 2010.

[146] Felix J. Villanueva. *Despliege eficiente de entornos inteligentes basado en redes inalambricas*. PhD thesis, University of Castilla-La Mancha, Spain, December 2009.

[147] Alexander Waibel and Rainer Stiefelhagen. *Computers in the Human Interaction Loop*. Springer Publishing Company, Incorporated, 2009.

[148] Mark Weiser. The computer for the 21st century. pages 933–940, 1995.

[149] Daniel Wigdor, Yuri Ivanov, and Christopher R. Wren. Soda pop zombies: soft drink consumption and motion. In *Proceedings of the 2007 workshop on Massive datasets*, MD '07, pages 8–9, New York, NY, USA, 2007. ACM.

[150] Robert Wilensky. *Planning and Understanding*. Addison-Wesley, Reading, MA, 1983.

[151] David E. Wilkins and Marie desJardins. A call for knowledge-based planning. *AI Magazine*, 22(1):99–115, 2001.

[152] Michael Witbrock, David Baxter, Jon Curtis, Dave Schneider, Robert Kahlert, Pierluigi Miraglia, Peter Wagner, Kathy Panton, Gavin Matthews, and Amanda Vizedom. An Interactive Dialogue System for Knowledge Acquisition in Cyc. In *In Proceedings of the Workshop on Mixed-Initiative Intelligent Systems. IJCAI*, pages 138–145, 2003.

[153] Christopher R. Wren, Yuri A. Ivanov, Darren Leigh, and Jonathan Westhues. The MERL motion detector dataset. In *Proceedings of the 2007 workshop on Massive datasets*, MD '07, pages 10–14, New York, NY, USA, 2007. ACM.

[154] Christopher Richard Wren, Yuri A. Ivanov, Ishwinder Kaur, Darren Leigh, and Jonathan Westhues. SocialMotion: Measuring the Hidden Social Life of a Building. In *Location- and Context-Awareness*, pages 85–102, 2007.

[155] David Wright, Serge Gutwirth, Michael Friedewald, Elena Vildjiounaite, and Yves Punie. *Safeguards in a World of Ambient Intelligence (The International Library of Ethics, Law and Technology)*. 1 edition.

Part VI

Appendix

Appendix A

Prototype Implementation

"Axioms in philosophy are not axioms until they are proved upon our pulses: We read fine things but never feel them to the full until we have gone the same steps as the Author."

– *John Keats*

Summary – *This appendix describes the details of the prototype implementation used to evaluate end user's degree of satisfaction.*

A.1 Introduction

Previous chapters have been establishing the theoretical basis of an architectural solution especially devoted to dealing with unexpected situations in Ambient Intelligence environments. The problem has been initially targeted from a dual perspective, that of the problem of understanding the situations that take place in the environment and the problem of devising the most appropriate way of reacting to a certain situation that requires from a system response. However, the proposed solution has only been described from a theoretical point of view and, nothing has yet been said about its feasibility and goodness. This appendix is devoted to addressing these aspects by building a prototype implementation and, although the features it offers are reduced, they are representative enough to draw important conclusions from the retrieved results.

The focus of this appendix is not the technology used but rather how it has been adapted to implement the proposed approach. For that reason, the prototype implementation is described here by stating the technological solutions that better fulfill the requirements of the theoretical approach. However, the use of the cited solutions is not mandatory and similar ones can be employed whenever the capabilities provided agree with the required ones.

The resultant prototype has been used to run the experimental validation tests intended to demonstrate the suitability of the proposed system for managing Ambient Intelligence environments. Specifically, the ultimate purpose of building a prototype has been to evaluate the system response whenever unforeseen events take place. Recall that one of the most important contributions of this thesis is the incorporation of common sense as the enabling key for generating system responses that have not been explicitly stated, but rather worked out on the basis of the surrounding context and the global goals pursued. These features need to be evaluated in the prototype system described in this chapter.

The best way to illustrate the insights of a prototype implementation is to describe how it behaves in a certain situation or case scenario. In this regard, the scenario in Figure A.1 depicts a typical

Ambient Intelligence environment, in which devices are devoted to monitoring the security conditions of the building, in order to foresee or prevent accidents or subversive actions from taking place, and to diagnose security faults when they occur and self-heal them. These devices are presence, sound, light, and proximity sensors, video cameras, and microphones, among the most relevant. The distributed architecture upon which these devices are deployed retrieves information from them and uses it to provide services such as object tracking, face recognition, geographical location, etc.

This comprehensive scenario of devices and services provides an appropriate context in which to test the suitability of the responses generated by the system, and a batch of different events and situations, generated by a synthetic data generator[1], is employed to this end.

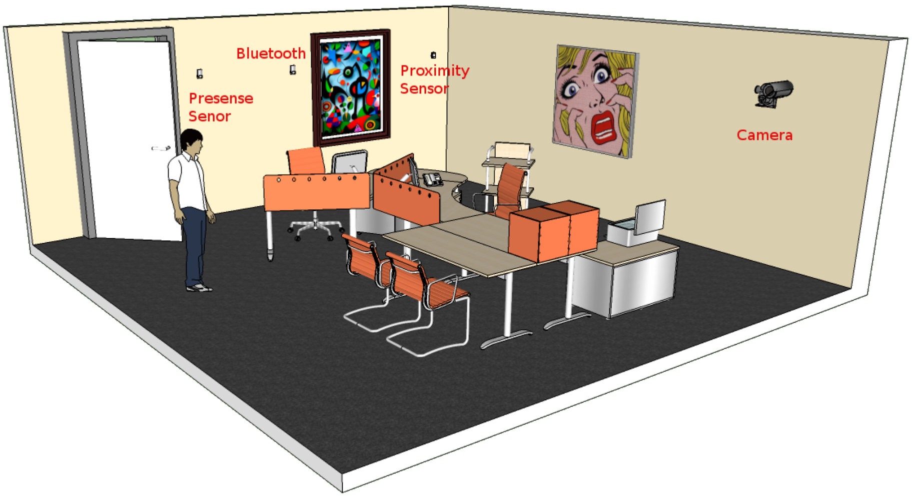


Figure A.1: Scenario representation

This appendix is organized as follows. First, the communication aspects of the prototype are addressed by resorting to a middleware strategy. In this sense, a solution based on the ZeroC ICE middleware technology is employed. Second, a Multi-Agent System is required for supervising the environment and for undertaking those environmental responses that better suit the current state of affair. To this end, a Jadex Framework has been used to support the implementation and management of a Multi-Agent system. The following section concerns the common-sense knowledge-base. In this sense, Scone is used not only to hold common-sense knowledge, but also to perform some basic inference and reasoning tasks upon such knowledge. Finally, the following section is concerned with the planning requirements. In this sense, a planning algorithm has been implemented as part of the Multi-Agent System. Finally, the most relevant conclusions of this chapter are presented in Section A.6

---

[1] A Synthetic Data Generator is available at: http://ailab.eecs.wsu.edu/casas/datasets.html

A.2 Middleware

The majority of the Ambient Intelligence systems can also be described as distributed heterogeneous applications, characterized by encompassing many different hardware devices with different protocols, operating systems, or implementation languages. In this sense, these types of applications strongly depend on communication mechanisms that ease and support interaction among themselves, otherwise preventing interaction from taking place in a seamless manner. Middleware technology is that type of communication mechanism devised for simplifying the procedures required for achieving an effective communication betwee heterogeneous and distributed devices.

Among the different strategies that can be found in the literature for implementing distributed applications, the object-oriented approach is one of the most successful solutions. The fact that every component of the distributed application is considered under this perspective as a distributed object provides an efficient and transparent mechanism to support their communication routines.

In this sense, the prototype implementation described here uses middleware technology as a way to overcome the difficulties related to the distributed and heterogeneous aspects of an Ambient Intelligence system. The middleware module is therefore one of the key elements of this architecture, essentially because it supports the connectivity of services and devices that run on different platforms, or using different network protocols, even when different programming languages are involved.

Among the different commercial implementations of distributed object-oriented middlewares, this prototype implementation uses ZeroC ICE[2]. ZeroC ICE is an object-oriented and CORBA-like middleware technology that provides the means (tools, API, libraries) to easily build object-oriented client-server applications.

Despite being similar in concept to CORBA, there are some additional resources that make ZeroC ICE the most appropriate technology for the prototype implementation devised here. In this sense, it is worth mentioning two of the most useful services provided by this technology, `IceGrid` and `IceStorm`, because of the important role they play in implementing this prototype. The `IceGrid` service provides a centralized manner for distributing applications. Additionally, the ZeroC ICE technology provides a mechanism, called `IceStorm`, that abstracts the details of implementing a publish/subscribe architecture. Event or communication channels can be easily implemented and managed, which is especially important for the Multi-Agent System, since it provides agents with the mechanisms for message exchange almost straightforwardly, by simply publishing the message to one of these channels. Agents subscribed to these channels automatically receive the message. For the sake of compatibility the agent messages adopt the FIPA-ACL standard [46].

Regarding scalability, the ZeroC ICE technology provides an implementation of the evictor pattern, as well as mechanisms to automate object persistence, that ensure the scalability of the system.

Certain ZeroC ICE concepts need to be settled before further details can be provided, although they might sound familiar to those readers who have had previous experience with object-oriented middlewares, such as CORBA. For example, a ZeroC ICE object, from now on referred to just as *object*, is no more than a conceptual abstraction that responds to client requests. An object has one or more interfaces, such that an interface is understood as the set of named operations provided by the object. Moreover, a single object can be instantiated in one or several servers. Each object has a unique object identity which is used to differentiate objects from each other. And finally, the *proxy* concept is used from the client side to contact a specific object. In order to invoke an operation on an object, the client invokes that operation on the proxy, whose address is already known by the client. From the client's point of view, the proxy apparently adopts the role of a local object that serves the

---

[2] http://www.zeroc.com/

invocation. However, in reality the ZeroC ICE run time, which runs in the background, is in charge of locating the ZeroC ICE object referred to by the proxy, activating it, transmitting the arguments, waiting for the operation to be completed and returning the operation results, if any. Bear in mind that the indirect proxy abstraction and the use of the same naming policy for interfaces is the cornerstone of automatic service composition. These two features basically standardize the method instantiation strategy. There is no need to know much about how to instantiate a method in a given service, since the indirect proxy and the implemented naming policy suffices to the perform this task.

Once the most relevant concepts of the ZeroC ICE middleware technology have been stated, it is possible to describe the most technical aspects of middleware prototype. First, recalling the role of the semantic model as the linking element, it seems obvious that this has to be incorporated into the middleware framework. Complying with the object-oriented facet of the middleware technology, every concept of the semantic model is mapped onto an object-oriented interface, whose methods correspond to the relationships established among the concepts of the semantic model. In this sense, the following code listing describes how the semantic model has been incorporated into the middleware framework.

```
module SemanticModel {
  dictionary <string, Object> Properties;

  struct Event {
    Properties theProperties;
  };

   interface EventMonitor{
     void report(Event e);
   };

  interface Device{
    void deviceName(string name);
    void setProperties(Properties spec);
    Properties getProperties();
  };

  sequence<Event>  EventSeq;

  interface Service {
   EventSeq performsAction(string idAction, Object thg);
   };
};
```

The *Service* concept adopts the shape of the `Service` interface. The *Action* concept of the model finds its equivalent in the `Action` interface implemented by all the methods that can be instantiated in the available services. Moreover, the semantic model notion of *Object* is mapped to the `Thing` interface. With regard to the functions formalized in the semantic model, the $PA(a,o)$ is implemented in the middleware layer by the `performAction(Action act, Thing thg)` method of the `Service` interface. The `agentOf(Action act)` method is the middleware implementation of the semantic model function $AO(a)$. Finally, the semantic model notion of *Event* is mapped to the `Event` structure.

It should be noted that the notion of *Context* does not have an equivalent in the middleware layer since the notion of context is irrelevant at this level. Moreover, there are other methods and interfaces that are not part of the semantic model although they are implemented in the middleware framework. For example the `setAction` and `getAction` methods are not part of the semantic model and their presence in the middleware layer responds to implementational needs.

With regard to the naming policy, every service in the system implements the `Service` interface. As a direct consequence of implementing this interface, every service in the architecture provides a common set of operations, such as the `performsAction` method, among others. Independently of the specific functionality provided by each service, there is a set of methods that are common to all services, since they all share a common naming policy.

Two main tasks can be assigned to the middleware layer. On the one hand it takes care of all the communication issues, abstracting and making transparent how services communicate, and on the other hand, it is responsible from instantiating each of the basic services comprising the composite service that has been devised by the Behavioral Response Generation module. This last responsibility is basically rooted in the naming policy, which is an indirect consequence of having implemented a common semantic model for actions and events. In this sense, the planning algorithm returns a list of actions, along with the services that provide them and the objects upon which they have to be performed. These objects can also be the results obtained from the previous action. For example, the action of detecting a face has to be performed on the result of the action that takes a snapshot.

The workflow of actions and objects upon which to perform them is therefore stated in a recursive manner, making the most of the object-oriented approach adopted by the middleware layer. Since everything under the object-oriented approach is considered to be an object, the result of undertaking a certain action provided by a service, can also be treated as an object, and therefore can also be provided with an action to be performed on it. However, since the role of instantiating the composite service is shared with the Multi-Agent System, further details on this matter will be provided in the next section.

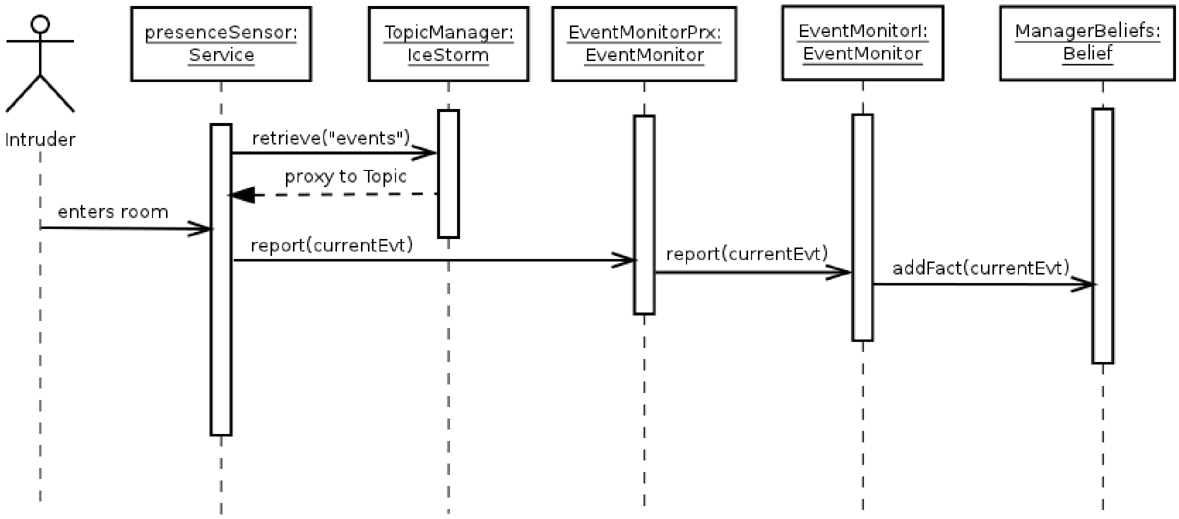


Figure A.2: Sequence diagram for the case scenario from the perspective of the middleware layer

Figure A.2 depicts the sequence diagram for the case scenario described in Section A.1. When the "*potential* intruder" enters the room, the presence sensor activates. This activation leads the associated service to publish a message to the channel labeled "*events*". This is a basic procedure that mainly consists in querying the `Topic Manager` about the proxy to the channel in which the message is to be published. On the other side of the communication channel the “`EventMonitor`“ has subscribed to the same channel so as to receive all the messages published there. The `EventMonitor` therefore receives the notification that is directly transferred to the `Manager agent` which, on the basis of the stated goals, will decide how to proceed.

A.3 Multi-Agent System

The role of the middleware framework is to take care of all the communication aspects involved in the construction of the prototype. Similarly, it is necessary to resort to an additional mechanism that supervises the events that occur in a supervised environment as well as the procedures that determine that a system response is required. Additionally, it cannot be forgotten that the response generation process is based on a goal-oriented approach, and therefore, it is also necessary to determine who is going to be responsible for adjusting the system behavior to the environmental goals, and how it will be done. It seems obvious that the best way to address these requirements is by means of a Belief, Desire, and Intention (BDI) Multi-Agent System.

The Multi-Agent System (MAS) works as a link between the Ambient Intelligence environment and the other elements of which the Ambient Intelligence framework is composed. Recall that the MAS is basically in charge of selecting the most appropriate plan given the current circumstances, and undertaking it with the available services. However, the MAS is part of a more global system, and therefore connectivity links need to be settled in order for those modules to interact. In this sense, the interaction with the other architectural elements is, once again, supported in the semantic model.

At the MAS level, the semantic model is incorporated by means of the Agent Communication Language (ACL) messages that agents can exchange among themselves. These ACL messages can be provided to the MAS by means of an OWL ontology, and therefore, the semantic model can be translated into an OWL ontology and provided to the MAS.

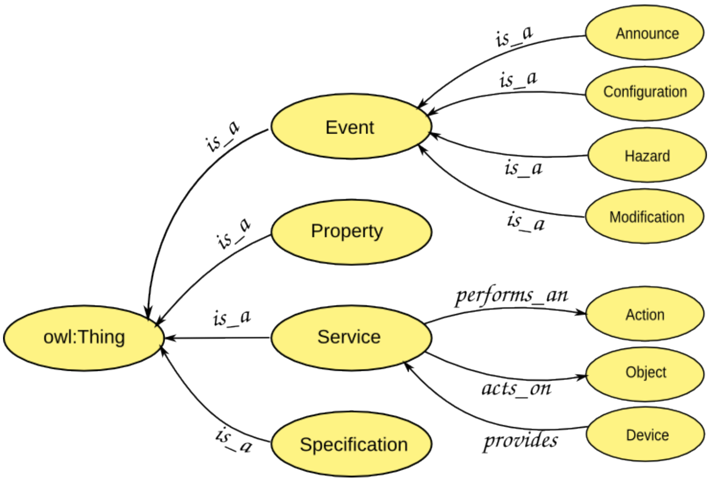


Figure A.3: Semantic model ontology for Ambient Intelligence

Figure A.3 depicts the OWL ontology mapping for the proposed semantic model. As occurred in the previous section with regard to the semantic model implementation at the middleware plane, there are some ontology classes that do not have an equivalent concept in the semantic model. The explanation for this is based on by the need to abstract implementation issues from the cognitive aspects of the model. For example, the taxonomy of event types is intended to adapt the MAS response to the specific type of event detected. Nevertheless, this aspect must remain specific to the MAS, since neither the middleware nor the planner need to know about the differences between event types.

The implementation of the goal-driven MAS has been accomplished here by means of a JADEX[3] framework, with some modifications to allow it to run upon the middleware layer. The JADEX framework supports the development of BDI (belief, desire, and intention) agents, understanding "*beliefs*" as the properties that an agent considers to be true, "*goals*" as the properties that an agent desires to be true, and finally "*plans*" as the actions that lead an agent to a desired goal. These basic instances define what is known as the agent's mental state.

The agent's beliefs in combination with contextual information (held in the Scone knowledge base) are what lead the agent's behavior towards the goals that the agent desires to achieve or maintain. Interaction between agents, the knowledge base, services and devices is based on the fact that all of them share the same semantic model.

For example, consider the aforementioned security surveillance context, and the occurrence of an event consisting of a presence sensor activation announcement coming from a restricted area. The following mental state is held by the agent supervising the context:

Belief(a, b): Agent *a* believes that an intrusion event b has taken place.
Goal(a, g): Agent *a* desires to halt the intruder.
Plan(a, p); Agent *a* resorts to a set of actions so as to halt the intruder.

However, how is this presence sensor event translated into an intrusive situation? How does the agent know which set of actions to perform when trying to halt an intruder if this plan depends on the place in which the event took place, and the resources available at the exact intruder location? These questions pose two of the main challenges facing Ambient Intelligence, namely, context understanding and autonomous and automatic behavior generation. Providing a solution to these two challenges is one of the main benefits of the research work undertaken here.

Figure A.4 depicts the different agents comprising the MAS, as well as the messages they exchange and the actions they undertake for the case scenario described in section A.1.

The main activity of the so-called `Manager agent` is the supervision of the events that occur in the supervised environment. To this end, the agent is subscribed to those communication channels in which the events are notified by the sensor devices that notice them. In this particular situation, the `event channel` is where the sensor publishes the notification and from which the `Manager agent` has received the notification. Since the notification comes from a specific sensor known to be deployed in a restricted access area, the sensor activation is interpreted as a possible unauthorized presence, and for that reason the agent labels this event as an *unauthorized presence event*, and notifies the `Perceptor agent` of this occurrence. Whenever the `Perceptor agent` believes that an unauthorized presence event has taken place, one of the goals that it triggers is intended to achieve intruder identification. Figure A.5 depicts the sequence diagram for this case scenario.

The following code, extracted from the goal description section of the `Perception agent`, shows how the `intruder_identification` goal is dispatched whenever an `eventType` occurrence is known to the agent or, in other words, when a fact of the `eventType` class is asserted to the agent's beliefs.

```
<!-- 02. Intruder identification -->
```

---

[3] Java based and FIPA compliant agent environment.

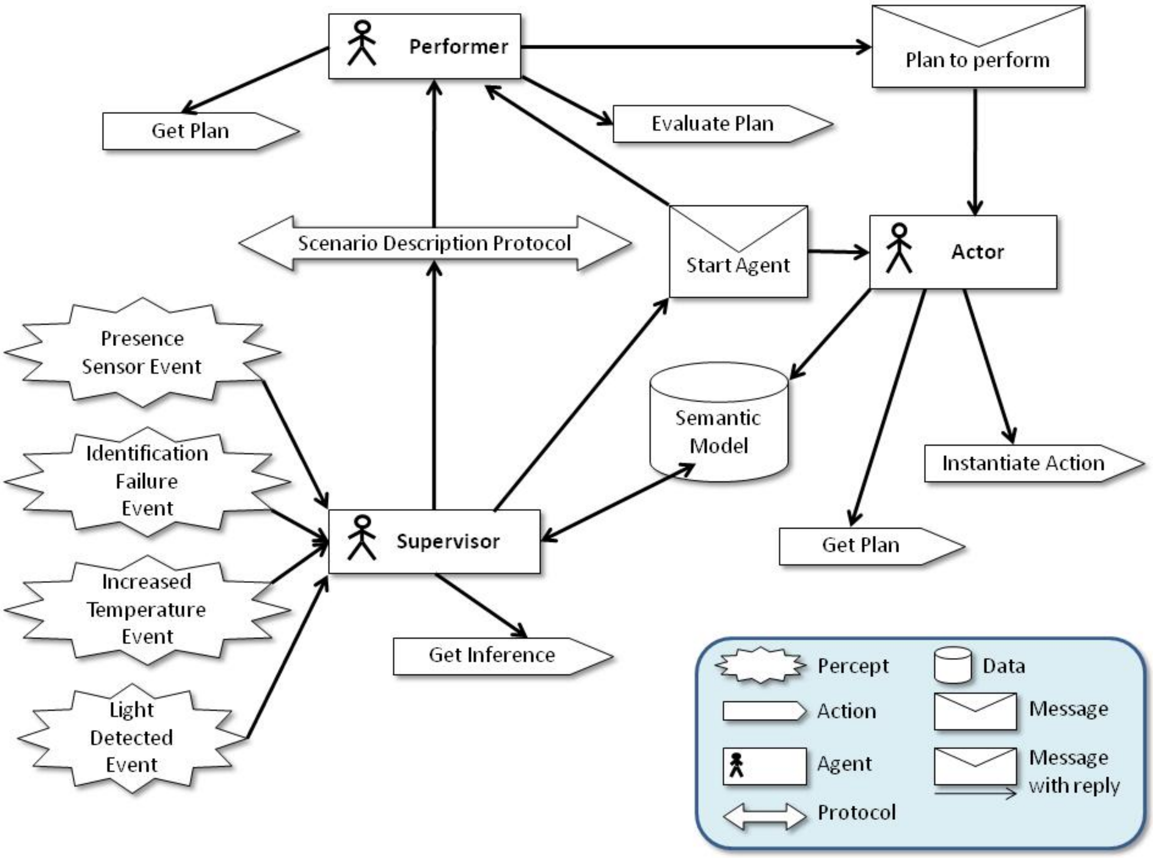


Figure A.4: Multi-Agent System overview diagram

```
  <achievegoal name="intruder_identification">
      <parameter name="unauthorisedPresence" class="Event">
          <bindingoptions>$beliefbase.eventTypes</bindingoptions>
      </parameter>
      <unique/>
      <!-- Create a new goal when new unauthorisedPresence
      event has been notified from the presence sensor. -->
  </achievegoal>
```

The `intruder_identification` goal requires a plan in order to be achieved. There are several ways in which to accomplish an intruder identification, one of which is by performing a biometric identification (fingerprints, iris, face recognition, etc.).

```
<!-- Plan intended to accomplish an intruder biometric identification -->
   <plan name="get_biometric_ID">
     <body class="GetBiometricIDPlan"/>
     <trigger>
       <goal ref="intruder_identification"/>
     </trigger>
   </plan>
```

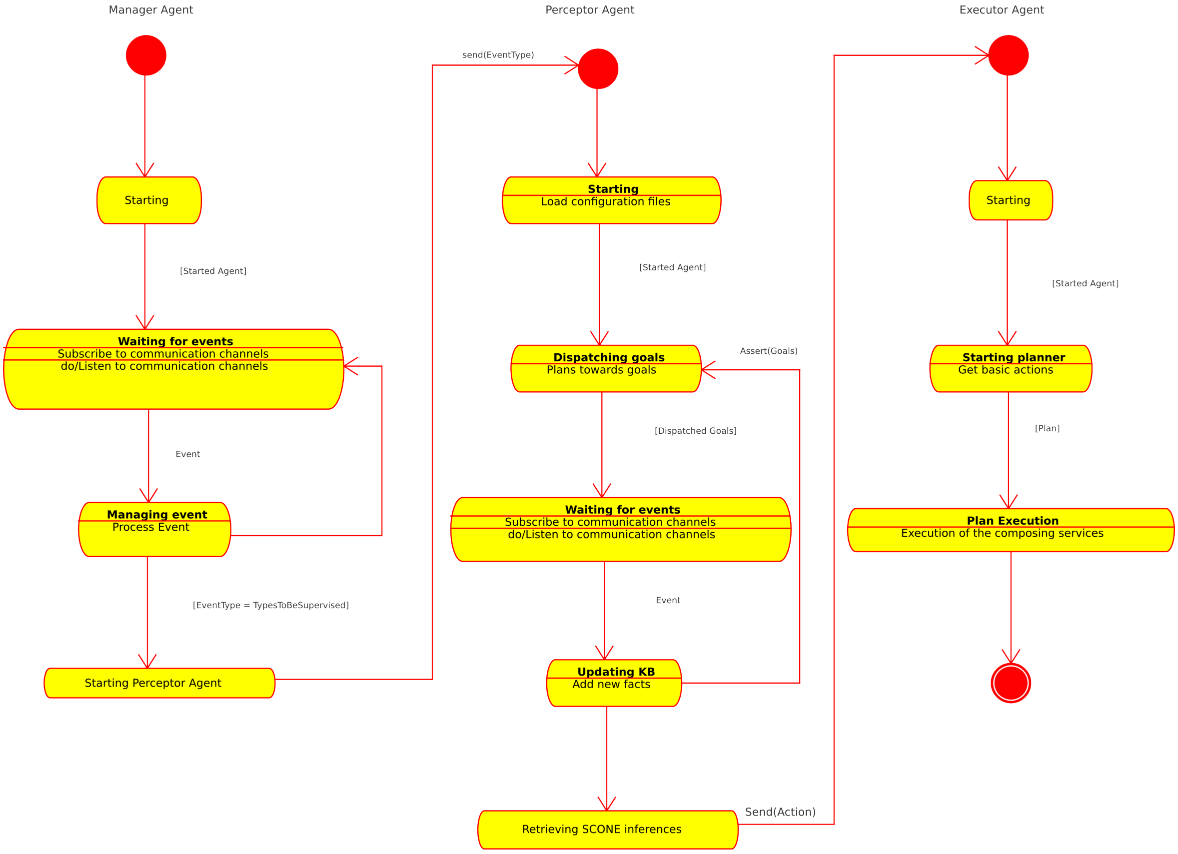


Figure A.5: State diagram for the Multi-Agent System

Plans in JADEX are traditionally static procedural recipes coded in Java. Constraining a plan to a static set of actions prevents the architecture from achieving the versatility and dynamism demanded by Ambient Intelligence. Rather than providing a static set of plans, these are provided dynamically and in an *ad-hoc* manner by resorting to a planning algorithm that identifies the course of actions that best fulfills the desired goals. Note the abstract character of the goal that gives the planner the responsibility for specifying the type of biometric identification that has to be carried out. As listed in the code shown below, the plan request specifies very general constraints, and it is simply engaged in accomplishing an `identification` action upon a `biometric feature` in order to obtain a `person identity` result.

```
....
 public void body()
 {
   // p = (P, A, O, R)
   List P = new ArrayList<List >();
   Planning pa = new Planning();
   P = pa.getPlan(P, "{identification}", "{biometric feature}","{person identity
       }");
  ....
  }
```

The result of the planning algorithm, stated as a set of quaternary elements of the form `<proxy, action, thing, result>`, is sent to the `Processor agent` which simply executes the given `action`, served at a certain `proxy`, upon the given `thing` in order to obtain a specific `result`.

```
....
//P has the result of executing the getPlan method. It was
// defined as List P = new ArrayList<List >();
Result rst;
Action act;
Thing thg;
String pr;
....
for(s=0; s< P.size(); s++){
  //Extracts the proxy string
  pr=extractString((String)((List)((P.get(s)))).get(1));
  //Extracts the action
  act=extractString((String)((List)((P.get(s)))).get(2));
  //Extracts the thing or assign it the result of the previous action.
  thg=extractString((String)((List)((P.get(s)))).get(3));
  if(thn.contains(''result of''))
     thn = rst;

  // get the proxy to the service associated to the action
  Ice.ObjectPrx base = ic.stringToProxy(pr);
  ServicePrx srv = ServicePrxHelper.checkedCast(base);
  rst = srv.performAction(act , thg);
  ...
}
```

The set of quaternary elements of which the plan is made up provides the MAS with the information required to automatically undertake the plan. Note that the agent plan has been composed in an *ad-hoc* manner, considering the availability of services and devices. Once again, it is important to highlight that the MAS capability to undertake plans generated on-the-fly is a direct consequence of using a common naming strategy for interfaces.

In order to carry out the proposed plan, the MAS simply invokes the `performsAction` operation on the service identified by the given `proxy` so as to perform the `action` upon the specified `thing`. Note how all this information is extracted from the quaternary set returned by the planner. Figure A.6 depicts the logic schema for the invocation method.

A.4 Knowledge-Base system

Doug Lenat, Marvin Minsky and Allan Newell have extensively and successfully discussed the bottleneck of intelligent systems. One of the main conclusions that can be drawn from their works is that common sense is indispensable to automating human-like behavior. Indeed, unpredicted situations can hardly be managed if common-sense knowledge has not been taken into consideration. For that reason, the Scone Knowledge Base has been selected to assume responsibility for holding common-sense knowledge and performing some reasoning tasks with this knowledge. The use of Scone is based on the need for common-sense knowledge modeling and reasoning capabilities, particularly when that knowledge refers to actions and events. As with the previously described modules, the semantic model has also been mapped into Scone.

In fact, the *Context* concept is one of the features of Scone that makes it so suitable for reasoning about actions and events. Refer to [22] for a description of the multiple context insights.

```
Ice::CommunicatorPtr ic;
Ice::ObjectPrx base = ic->stringToProxy(Proxy);
ServicePrx srv = ServicePrx::checkedCast(base);
Result r = srv->performAction(action, thing);
```

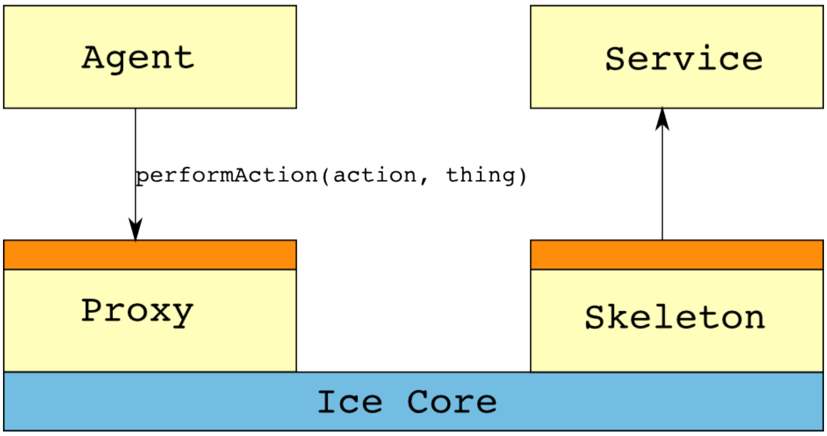


Figure A.6: Logic schemata for remote method invocation

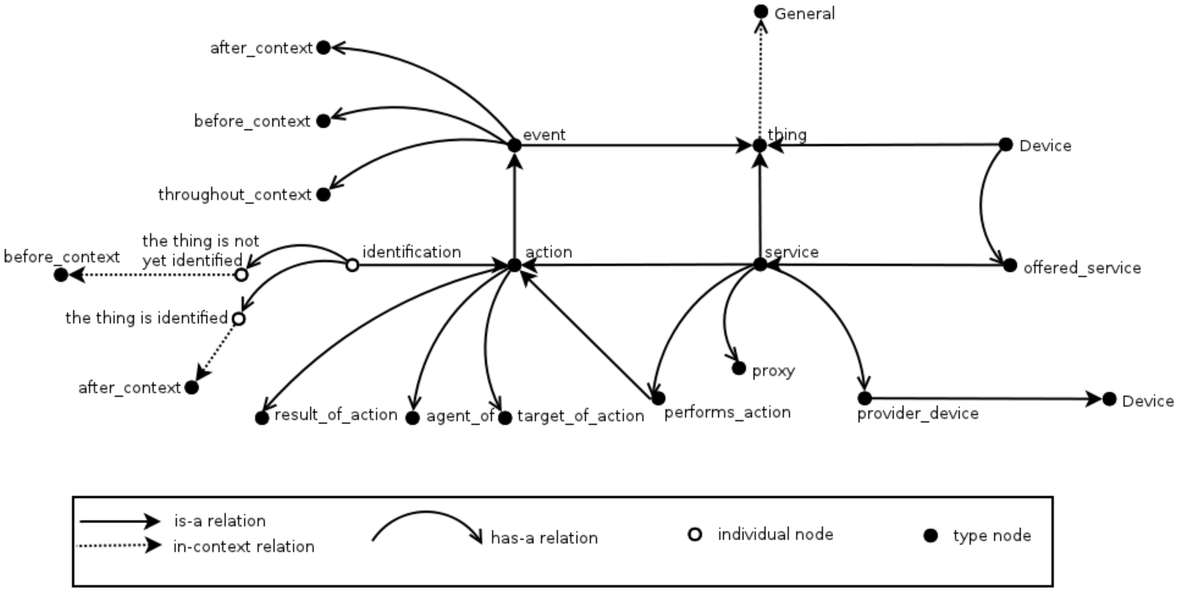


Figure A.7: Semantic Model in Scone

Moreover, not only contexts are relevant for the modeling of actions and events, but also the services that provide them, the agents that bring them up, or the generated outputs. Figure A.7 depicts how the semantic model has been mapped onto the Scone KB. Note how the semantic model concepts and relationships are implemented, respectively, as nodes and links in Scone. This semantic model has been used as a foundation for the coding of a dictionary of actions and events.

The following code listing states how the semantic model is implemented into the Knowledge-Base:

```
(new-type {event} {thing})
(new-type {service} {thing})
(new-type {device} {thing})
(new-event-type {action} '({event})
                :roles
                ((:type {target-of-action} {thing})
                 (:indv {result} {thing})))

(new-type-role {performs-action} {service} {action})
(new-type-role {agent-of} {action} {service})
(new-type-role {offered-service} {device} {service})
(new-type-role {provider-device} {service} {device})
(new-type-role {result-of-action} {event} {thing})
(new-type-role {object-of} {event} {thing})
```

Roles associate properties to the concepts that hold them. For example, the role `performs-action` associated to the concept of `service` is used to state that the typical service has at least one action as the action performed by that service.

```
(the-x-of-y-is-z {performs-action} {presence-detection-service}
  {presence-detection-action})
```

The previous statements associates the `presence-detection-action` to the `presence-detection-service`. This statement can be therefore interpreted such that the `presence-detection-service` performs the `presence-detection-action`.

The following code listing shows the representation, using the Scone language, of the `capture` and the `capturingImage` events. It is worth noticing the second is a specialization of the first, and therefore, the content of the respective contexts is inherited. For example, the `before context` of the `capturingImage` event inherits from the `capture` event the fact that the object has not yet being captured. Since the `captionObject` adopts the shape of an `instant photo frame` and the `captionSource` that of the `light photons`, in other words, it can be said that the `light photons` have not yet been captured into an `instant photo frame`.

```
(new-event-type {capture} '({event})
        :roles
        ((:indv {captionSource} {thing})
         (:indv {captionObject} {thing})
         (:indv {captionTarget} {data}))

  :throughout
  ((new-statement {captionObject} {is noticed in} {captionSource}))
  :before
  ((new-not-statement {captionObject} {is recorded in}
        {captionTarget}))
  :after
  ((new-statement {captionObject} {is recorded in}
        {captionTarget})))

(new-event-type {capturingImage} '({capture} {action})
```

```
:throughout
((the-x-of-y-is-a-z  {captionSource} {capturingImage}
      {light photons})
 (the-x-of-y-is-a-z  {captionObject}
      {capturingImage}{instant photo frame})
 (the-x-of-y-is-a-z  {captionTarget}  {capturingImage}
      {imageFile})))
:after
((new-statement {imageFile} {is picture of} {instant photo frame})))
```

For example, the above lines describe (from a common-sense perspective) what the `capture` event represents in terms of relevant elements and states of the world involved (*before*, *throughout*, and *after contexts*). Event roles symbolize those domain elements that characterize the world states. For example, the `captionSource` role is played by the thing being captured. When referring to the `capturingImage` action, the `captionSource` role is specified in the light photons captured by a photographic camera. The after context for the `capturingImage` action describes a state of the world in which, after it takes place, the action results in a new state in which there is an image file, picturing the instant photo frame captured by the camera.

The planning algorithm, based on the actions and events dictionary and the domain knowledge held in the Scone KB, resorts to the inference capabilities of Scone to devise the course of actions which, given a desired state of the world, lead to its realization.

The following lines show Scone's strengths with regard to inferring and deducing the knowledge that seems obvious to people, but is so difficult for computers to handle. The Scone type and property hierarchy KB and its implementation of the marker-passing inference strategy, provide the means to enhance planning with common-sense knowledge and reasoning capacity, resembling the process by which people make decisions.

For example, when attempting to figure out the `identity` of an intruder by performing the `identification` of a `biometric feature`, the first step consists of determining the existence of a service that is capable of providing such functionality. At first glance, one might easily conclude that this is too generic a task to be provided by a service, and Scone is no exception. When asked about the existence of such a service, Scone answers that there is no type or individual node whose `performs-action` role is the `identification` event. In other words, the `identification` event is not directly provided by any of the available services:

```
CL-USER> (x-is-the-y-of-what? {identification}  {performs-action})
{identification} is not known to play the {performs-action} role of anything.
NIL
```

At this stage, a sensible approach is to seek those events or actions that cause the same effects as those caused by the `identification` event:

```
CL-USER> (list-events-causing-x (new-statement {biometric feature}
(car (list-parents(car (list-after {identification})))) {person identity}))
({recognition} {faceRecognition}
 {identityIdentificationAccess} {identityIdentification})
```

The Scone answer to this query is a set of actions and events that produce the same effects as the `identification` event. However, not all of them are equally useful, and those directly provided by available services are preferred to those that cannot be served by available services. In order to figure this aspect out, Scone is again queried about the existence of services performing the given actions.

As listed below, the `recognition` action is not performed by any of the available services, while the `faceRecognition` action is indeed provided by the cited service:

```
CL-USER> (x-is-the-y-of-what?  {recognition} {performs-action})
{recognition} is not known to play the {performs-action} role of anything.
NIL

CL-USER>  (x-is-the-y-of-what?  {faceRecognition} {performs-action})
{SimpleRecognizer:default —p 12000}
```

Note that the `performs-action` property (the so-called role) symbolizes the action or set of actions capable of being undertaken by individual nodes of the `service` type node.

When queried about the existence of an individual `service` performing the action of `faceRecognition` Scone answers that the individual, with proxy property `SimpleRecognizer:default -p 12000`, is capable of performing an equivalent `identification` event. The proxy property is also a role or property of the `service` node. This is used to hold the remote location address from which actions can be called to be executed.

In order to match the request, not only must the *after contexts* be equivalent, but also the items upon which actions are performed. Therefore, it is also necessary to check that those items supporting the equivalent actions or events are equivalent. In other words, the following steps consist of checking that the `faceRecognition` action can be performed upon a `biometric feature` as stated in the initial requirements:

```
CL-USER> (list-all-x-of-y {object-of}  {faceRecognition})
({events:face})

CL-USER> (can-x-be-a-y? {face} {person identity})
T
```

`Face` is the item upon which the `faceRecognition` action is performed. It is obvious to people that a `face` is also a `biometric feature`, and this is confirmed by Scone when queried. Since the `face` object works as an input to the `faceRecognition` action, the following step consists of devising how to obtain or satisfy the action requirements:

```
CL-USER> (x-is-the-y-of-what? {faceRecognition} {performs-action})
{SimpleRecognizer:default —p 12000}

CL-USER> (list-events-preceding {faceRecognition})
({detectingFace})
```

If the `detectingFace` action is required so as to permit the `faceRecognition` action to take place, Scone should once again be queried about the inputs or requirements for the `detectingFace` action, and should also verify whether any of them is compliant with the `face` object. The following lines show how to implement such an interaction with Scone:

```
  CL-USER> (list-all-x-of-y {object-of} {detectingFace})
  ({captureResult of recordingImage}
  {A-role of is picture of})
  CL-USER> (can-x-be-a-y? {captureResult of recordingImage} {face})
  T
```

The interpretation of the above results concludes that the `detectingFace` action has to be performed either upon the result of a recording image device or a picture file. However, apart from the required input, the `detectingFace` action might also demand some other requirements to be undertaken. Scone is therefore queried about this matter:

```
CL-USER> (x-is-the-y-of-what? {detectingFace} {performs-action})
{SimpleDetector:default −p 11000}
CL-USER> (list-events-preceding {detectingFace})
({capturingFace} {performs-action} {recordingImage}
{recordingVideo})
CL-USER> (list-all-x-of-y {object-of} {capturingFace})
({captionTarget of capturingBiometricFeature}
{B-role of is recorded in}
{captureResult of recordingVideo)}
{captureResult of recordingImage}
{captureResult of detectingLight}
{captureResult of detectingPresence}
{A-role of is picture of (0−1290)})
```

Scone concludes that in order to fulfill the requirements demanded by the `detectingFace` action, the following could be undertaken: `capturingFace` `performs-action` `recordingImage` `recordingVideo`.

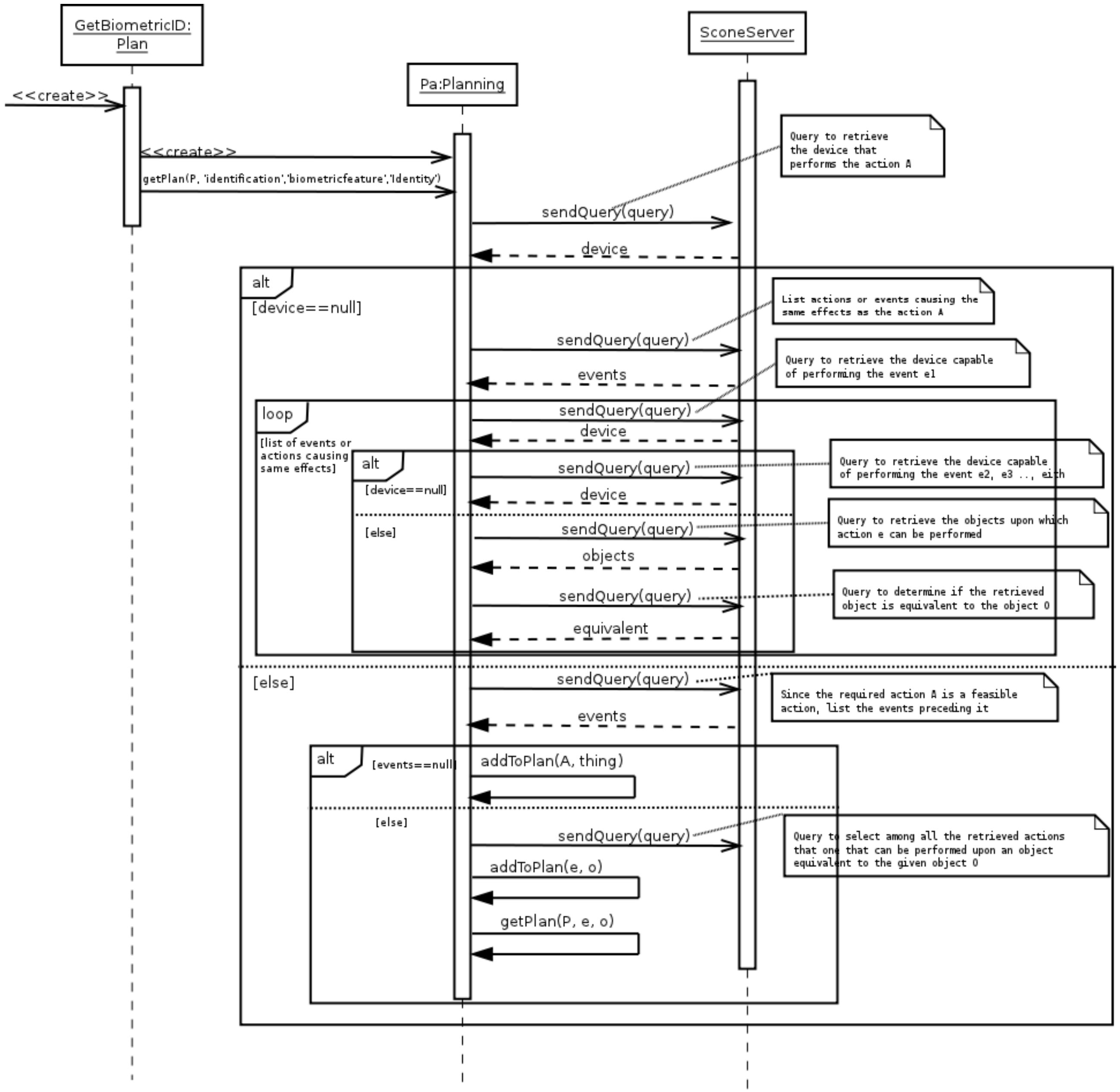
```
CL-USER> (can-x-be-a-y? {captionTarget of capturingBiometricFeature}
         {captureResult of recordingImage})
T

CL-USER>  (x-is-the-y-of-what? {capturingFace} {performs-action})
{videoCamera1Service}

CL-USER> (list-events-preceding {capturingFace})
NIL

CL-USER> (b-wire (car (list-after {capturingFace})))
{imageFile}
```

Steps are repeated using different actions until a point is reached at which the action does not require any inputs, and can therefore be directly accomplished. When this point is reached, Scone is asked about the result of the action. As can be observed in the above lines, the output of the `capturingFace` is an image file, from which a face can be detected in order to perform a face recognition action so as to determine the intruder's identification.

The planning algorithm proposed in this thesis is intended to automate the generation of the queries presented above. By starting from a ternary query composed of the action, the object or item that receives the action, and the expected result, the planning algorithm is able to find the course of actions that provides a similar functionality. To summarize, the result provided by the planner for the example analyzed here generates the following course of actions:

```
((capturingImage, thing, imageFile),
 (detectingFace, imageFile, imageFile),
 (faceRecognition, face, person identity),
 (identification, biometric feature, person identity))
```

A.5 Planner



Query to retrieve
the device that
performs the action A

List actions or events causing the
same effects as the action A

Query to retrieve the device capable
of performing the event e1

Query to retrieve the device capable
of performing the event e2, e3 .., eith

Query to retrieve the objects upon which
action e can be performed

Query to determine if the retrieved
object is equivalent to the object O

Since the required action A is a feasible
action, list the events preceding it

Query to select among all the retrieved actions
that one that can be performed upon an object
equivalent to the given object O

Figure A.8: Sequence diagram for the case scenario from the perspective of the planning algorithm

Making the most of service versatility enables Ambient Intelligence systems to respond to whatever the needs are by adapting available services and devices to the desired functionality. Indeed, in this context, arising needs are treated as a desire to perform actions upon objects. By making this assumption and adapting a Hierarchical Task Networks (HTN) approach to consider actions as tasks, the task of satisfying arising needs can be automatically accomplished by means of an HTN-like planner.

The actions that can be performed by the system, at a specific location and time, are determined by the devices and services available at that location and time. Those actions that cannot be performed, owing to the lack of services that provide such functionality, are named here as non-feasible actions. Whenever the system demands the execution of a non-feasible action, the planner comes into play.

Figure A.8 depicts the sequence diagram for the case scenario considered here, now from the perspective of the planning algorithm. As can be seen, the planning algorithms basically interacts with the Scone KB from which alternative options are analyzed in seeking those actions that produce the same effects as the `identification` action performed upon a `biometric feature`. Recall that the Scone KB not only holds common-sense knowledge about general terms, but it also holds information about the devices that are currently deployed in the supervised context, as well as additional information about them (properties, proxy, location, etc.).

The Scone KB system can also be deployed as a server, to which TCP connections can be established. The following code listing corresponds to the java implementation of the planning algorithm presented in Section 7.4.2.

```
public List getPlan(List P, String A, String O, String R){

        List E = new ArrayList();
        List Ob = new ArrayList();
        List p =  new ArrayList<String>();
        String query = "(x-is-the-y-of-what? "+A+"  {performs-action})";
        String answer, e=A, o=O;
        boolean found = false;
        String device = sendToScone(query);
        if(device.equals("NIL")){
            query =  "(list-events-causing-x (new-statement "+O+" (car
            (list-parents(car (list-after "+A+")))) "+R+"))\n";
            E = getListFrom(sendToScone(query));
            ListIterator e_i = E.listIterator();
            while(e_i.hasNext() && !found){
                e = (String)(e_i.next());
                query =  "(x-is-the-y-of-what? "+e+"  {performs-action})\n";
                if((sendToScone(query)).equals("NIL"))
                    e_i.remove();
                else{
                    query = "(list-all-x-of-y {object-of} "+e+")\n";
                    Ob = getListFrom(sendToScone(query));
                    ListIterator o_i = Ob.listIterator();
                    while(o_i.hasNext() && !found){
                        o = (String)(o_i.next());
                        query = "(can-x-be-a-y? "+o+" "+O+")\n";
                        if((sendToScone(query)).equals("T"))
                            found = true;
                    }
                }
```

```
            }

            query = "(b-wire (car (list-after "+e+")))";
            getPlan(P, e, o, sendToScone(query));

        }
        else{
            System.out.println("The device is: "+ device);
            query = "(list-events-preceding"+A+")";
            E = getListFrom(sendToScone(query));
            if(E.size() ==0){
                query = "(b-wire (car (list-after "+A+")))";
                p.add(A);
                p.add("{thing}");
                p.add(sendToScone(query));
                p.add(device);
                P.add(p);
            }
            else{
                ListIterator e_i = E.listIterator();
                while(e_i.hasNext()&& !found){
                    e = (String)(e_i.next());
                    query = "(is-x-a-y? "+e+"{action})";
                    if((sendToScone(query)).equals("T")){
                        query = "(list-all-x-of-y {object-of} "+e+")\n";
                        Ob = getListFrom(sendToScone(query));
                        ListIterator o_i = Ob.listIterator();
                        while(o_i.hasNext() && !found){
                            o = (String)(o_i.next());
                            query = "(can-x-be-a-y? "+o+" "+O+")\n";
                            if((sendToScone(query)).equals("T"))
                                found = true;
                        }
                    }
                }
                query = "(b-wire (car (list-after "+A+")))";
                p.add(A);
                p.add(o);
                p.add(sendToScone(query));
                p.add(device);
                P.add(p);
                getPlan(P, e, o, sendToScone(query));
            }

        }


        return P;
    }
```

A.6 Interim conclusions

This chapter is basically concerned with the implementation of a prototyped version of the architectural solution described in the previous chapters. Enough technical details have been provided so that the prototype implementation can be replicated.

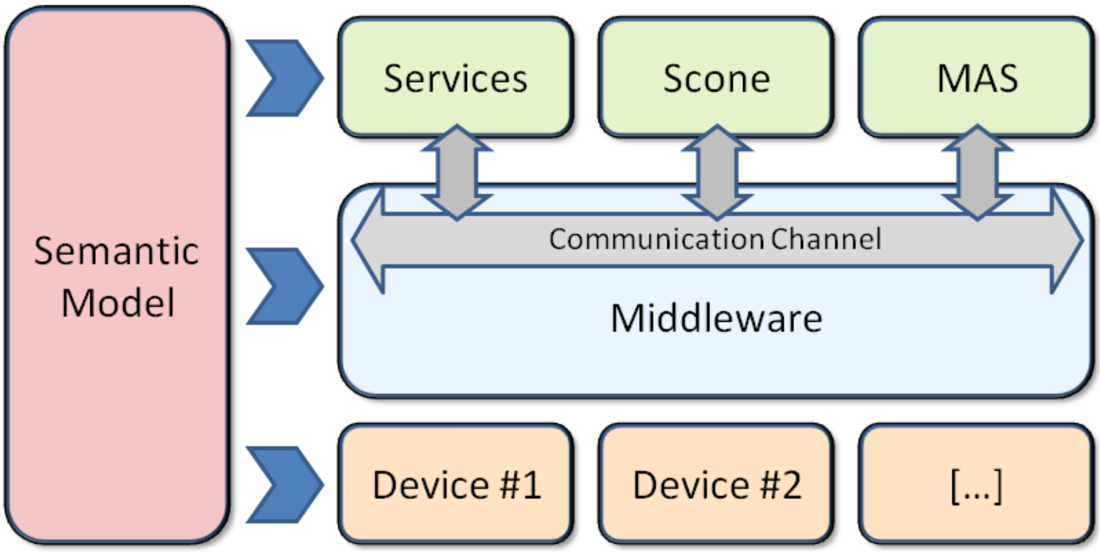


Figure A.9: System architecture overview.

This constructed prototype is basically composed of three modules:

- The middleware framework
- The Multi-Agent System (MAS)
- The common-sense system

Figure A.9 provides the reader with a comprehensive system overview from the perspective of the modules involved in the architecture. The use of the cited technologies is not mandatory, and similar ones can be employed if they fulfill the stated requirements.

Besides, the prototype implementation details can be used to assess how satisfied the users are with responses worked out by the system. In this sense, the evaluation methodology proposed in **??** can be used for this purpose. The system can come up with three different types of responses, namely: a) basic services, which are straightforwardly provided by devices; b) responses that encompass basic and composite services; c) and composite services. The proposed evaluation methodology can be used to assess which of these types of services best satisfies the user preferences. In this sense, the proposed methodology has been tested to evaluate the user response to a simulated case scenario. The test taken has led to the conclusion that users are more satisfied with composite services because they satisfy their preferences and requirements better.

The prototype system's capability to compose services, in such a way that they are more valuable than basic services, complies with one of the main aims of this research, as stated in Section 1.3. The leverage of a common-sense planning strategy for automatic service composition has therefore proved to be a successful approach for behavioral response implementations.

Este documento fue editado con *GNU Emacs*
y tipografiado con LaTeX en un sistema *GNU/Linux* a partir de la plantilla realizada por Francisco Moya Fernández para su tesis.
Ciudad Real, September 30, 2011.