Towards self-adaptive and cooperative multiagent systems for ambient assistive applications

Thèse de doctorat de l'Université Paris-Saclay et de l'université de Tunis

École doctorale n° 580, Sciences Et Technologies De L'information Et De La Communication (STIC)
Spécialité de doctorat : Informatique
Unité de recherche : Université Paris-Saclay, Univ Evry, IBISC, 91020, Evry-Courcouronnes, France
Référent : Université d'Évry Val d'Essonne

Thèse présentée et soutenue à Evry, le 17 décembre 2020, par

Amina MADDOURI

Composition du jury:

Salima Hassas Professeur, LIRIS, Université Claude Bernard Lyon 1	Présidente
Abderrazak Jemai Professeur, INSAT, Université de Carthage, Tunisie	Rapporteur et Examinateur
Amel Bouzeghoub Professeur, Télécom SudParis, Institut Mines-Télécom	Rapportrice et Examinatrice
Chantal Taconet Maître de Conférences, HDR à Telecom SudParis	Examinatrice
Meritxell Vinyals Ingénieur de recherche, Phd – Institut CEA LIST	Examinatrice
Samir Otmane Professeur, Université d'Evry, Université Paris-Saclay	Directeur de thèse
Jalel Akaïchi Professeur, ISG, Université de Tunis, Tunisie	Directeur de thèse
Nadia Abchiche-Mimouni Maitre de Conférences, Université d'Evry, Université Paris-Saclay	Co-encadrante

Thèse de doctorat

NNT: 2020UPASG035

Abstract

Our research investigates adaptive multiagents Systems in open environments. We are interested in an open, dynamic and unstable context in which agents can dynamically cooperate. Since the environment is dynamic, due to the availability or the lack of connected objects and to the diversity of situations, cooperation mechanisms among connected objects must be dynamically and adaptively designed. The adaptation takes into consideration the context of the person (elderly or sick people) as well as the existing interaction protocols.

The aim of this work is to propose a contribution to the automation of the design and the deployment of adaptive multiagent systems for ambient assistive applications. We propose a new approach for Generating Automatically an Adaptive Multiagent system for Ambient Assistive Applications (GAAMAAA). This approach focuses on the design and implementation of the interaction protocol ontology and the creation steps of the MAS. For this purpose, we define the agent-based architecture of GAAMAAA, we give a protocol ontology for protocol modeling and ambient assistive ontology to create agents. We specify an execution model of protocols that enables agents to dynamically take part into interaction. We carry out the dynamic integration of these protocols into agents involved in the corresponding interaction. Our approach is validated by implementing a localization robot scenario, which the GAAMAAA provides support for two most used interaction protocol ICNP and Coalition. The evaluation shows the effective performance of interaction protocol based on sensitivity analyses method.

Keywords: Multiagent Systems, Interaction, Ontology, Connected Objects, Automation , Adaptation.

Résumé

Nos recherches portent sur les systèmes multiagents adaptatifs en environnement ouvert. Nous nous intéressons à un contexte ouvert, dynamique et instable dans lequel les agents peuvent coopérer de manière dynamique. Comme l'environnement est dynamique, en raison de la disponibilité ou du manque d'objets connectés et de la diversité des situations, les mécanismes de coopération entre les objets connectés doivent être conçus de manière dynamique et adaptative. L'adaptation prend en considération le contexte de la personne (personnes âgées ou malades) ainsi que les protocoles d'interaction existants.

L'objectif de ce travail est de proposer une contribution à l'automatisation de la conception et du déploiement de systèmes multiagents adaptatifs pour les applications d'assistance ambiante. Nous proposons une nouvelle approche pour la génération automatique d'un système multiagent adaptatif pour les applications d'assistance ambiante (GAAMAAA). Cette approche se focalise sur la conception et la mise en œuvre de l'ontologie des protocoles d'interaction et sur les étapes de création du SMA. Pour cela, nous définissons l'architecture à base d'agents de GAAMAAA, nous donnons une ontologie de protocole pour la modélisation du protocole et une ontologie d'assistance ambiante pour créer des agents. Nous spécifions un modèle d'exécution des protocoles qui permet aux agents de prendre part à l'interaction de manière dynamique. Nous réalisons l'intégration dynamique de ces protocoles dans les agents impliqués dans l'interaction correspondante. Notre approche est validée par la mise en œuvre d'un scénario de localisation de robot, que le GAAMAAA prend en charge pour deux protocoles d'interaction les plus utilisés, ICNP et la coalition. L'évaluation réalisée montre la performance effective du protocole d'interaction basé sur la méthode des analyses de sensibilité.

Mots clés: Systèmes multiagents, Interaction, Ontologie, Objets connectés, Domotique, Adaptation.

Acronym

AA	**A**mbient **A** ssistance
AE	**A**mbient **E**nvironment
AmI	**A**mbient **I**ntelligence
BDI	**B** eliefs **D**esires **I**ntentions
CO	**C**onnected **O**bjects
Coalaa	**C**oalitions for **A**mbient **A**ssisted **L**iving **A**pplications
AI	**A**rtificial **I**ntelligence
MAS	**M**ulti**A**gent **S**ystems
OWL	**W**eb **O**ntology**L**anguage
DU	**D**egree **U**rgency
LP	**L**evel **P**recision
LI	**L**evel **I**ntrusion
AAL	**A**mbient **A**ssited **L**iving
GAAMAAA	**G**enerating **A**utomatically an **A**daptive **M**ultiagent systems for **A**mbient **A**ssistive **A**pplications

Acknowledgments

First and foremost, praises and thanks to the God, the Almighty, for His showers of blessings throughout my research work to complete the research successfully. Thanks God ...

I would like to express my profound gratitude to Dr, Nadia Abchiche-Mimouni, my co-supervisors, who started by proposing this PhD project and believing in my capacity to accomplish it. I am very grateful for her permanent support, scientific guidance, constant optimism and encouragement throughout this long walk with all its “ups and downs”. I deeply appreciate her careful revision of all my work.

I would like to express my deep and sincere gratitude to my supervisor, Pr. Samir Otmane, for giving me the opportunity to do research and providing invaluable guidance throughout this research. His dynamism, vision, sincerity and motivation have deeply inspired me.

I owe my deepest gratitude to my supervisor Pr. Jalel Akaichi for his supervision.

I would also like to thank all the staf of the IBSIC laboratory for providing me such a great work environment.

I am deeply indebted to Pr. Abderrazek Jemai for his support, encouragement and valuable discussions concerning this research.

I am extremely grateful to my parents for their love, prayers, caring and sacrifices for educating and preparing me for my future. Also I express my thanks to my sisters and brothers.

Finally, my thanks go to all the people who have supported me to complete the research work directly or indirectly.

Contents

List of Figures

List of Tables

List of Algorithms

Introduction

Currently, there is an important idea for the user in the computer science research; the integration of various technical applications in our traditional environments such as home, workplace, leisure centers and the city.
In the near future, these environments will recognize our tastes and our preferences, and they will help us achieving our goals and our needs. People will interact with an environment that will be aware of their presence and the general context and that can adapt and respond to their needs, habits and emotions. The ultimate goal is to get the comfort of the users in their daily lives and to respect the security and confidentiality of each of them without intrusion.
The environment contains heterogeneous devices that seem isolated but with the help of existing networks technologies, these devices can communicate and collaborate with each other and thus form a system called ambient system. This communication may be wireless like WiFi technology, Bluetooth or wired Ethernet.
The ambient system should supplement the needs of users without interference in the needs of other ones. Needs and tastes are important but then there is also an essential concept: assistance in situations where difficulties are encountered by the user such as elderly or sick persons. For example, the ambient system can help to find solutions to the situation of falling of Alzheimer sufferers or at least collect important information and thus help in decision making. Our application context is to assist an elderly or a sick person in loss of autonomy at home by providing assistive applications based on cooperation among a set of Connected Objects (COs). The context of such applications is highly dynamic and changing due to the COs that may or may not be available, but also due to the context related to the person (his/her habits, preferences, movements...). The presence of a mobile robot can help to deal with such dynamic situations. Indeed, a robot can interact and cooperate with COs to decline a set of services and teleservices in order to facilitate the daily life of the person and his entourage at home.

Multiagent Systems (MAS), initially introduced in the context of Distributed Artificial Intelligence, are currently at the intersection of several disciplines of computer science such as Software Engineering, Artificial Intelligence (AI), and Distributed Systems. They are very well recognized both in the research community and in the industry as a framework facilitating the development of complex and adaptive systems [Beaumont & Chaib-draa 2007]. They are dynamic, evolving and composed of autonomous and heterogeneous entities that act in cooperation to achieve a common objective. One of the most important and principal qualities of MAS is the communication between the agents representing a capital mechanism inside agents community. Ambient Intelligence (AmI) is an emerging discipline whose aim is to brings AI in our daily life, in particular for assisting elderly or sick persons. Unfortunately, the existing systems have not been as successful as expected, in particular because of their lack of flexibility and adaptation to people's environments and profiles. Indeed, the changing nature of networks of COs and the human nature which is unpredictable are problematic. This thesis addresses the problem of adaptation and self-adaptation for ambient assistive applications thanks to MAS.

Scientific challenges

The main addressed challenge in this work is the following:

Ensuring dynamic interactions and automatic agent generation in an open and dynamic environment such as ambient assistance, using the interaction protocols of a MAS.

In order to deel with such a challenge, we will try in this thesis to adresse the questions below:

→ How to ensure the matching between an ambient environment and a MAS?

→ Which protocol modelling to adopt?

→ How to deploy and run an interaction protocol dynamically?

→ How to ensure dynamic interactions and automatic agent generation in an open and dynamic environment?

→ How to ensure a self-adaptive MAS to ambient assistance application?

Aim and scope

This research focuses on improving adaptation and self-adaptation of MAS in open environment for ambient assistive applications. Let us briefly present below the main contributions of this thesis:

- An ontology of interaction protocols (PrOnto) inspired from [W.Bouaziz *et al.* 2009], [V.Tamma *et al.* 2005], [T.Jarraya & Z.Guessoum 2007], which is presented in chapter 4. This ontology is considered as a model for defining the structure shared by all the protocols. More precisely, this ontology makes it possible both to describe the profile of the protocols as well as their control structure. The profile defines what the protocol does, its purpose, specifying the protocol category, the problem types resolved by each protocol, and specifying some of its static properties. The profiles will be used to perform the dynamic selection of the protocols. The control structure defines its behavior and its functioning through the description of the actions that it supports and the scheduling of the messages that it allows exchanging.

- An architecture for Automatically Generating Adaptive Multiagent Systems for Ambient Assistive Applications (GAAMAAA)[Maddouri *et al.* 2019]. Exposed in chapter 4 which makes it possible to support the entire life cycle of protocols from their specification to their execution. GAAMAAA architecture is an attempt to automatically generate a set of cooperative agents in order to provide a service adapted to a dependent, elderly, or a suffering person maintained at home. The latter is equipped with a set of CO and possibly with a robot. This work is in the continuity of the COALAA project. It has led to:

- A system for generating agents; this system makes it possible to automate the creation of MAS from the Ambient Assistance (AA) ontology.
- An interaction protocol generation system; this system makes it possible to assign interaction protocols to the new agents created by the system, depending on the desired effect (robot location) requested by the requesting agent and updating system.

- A strategy presented in chapter 5 allowing an agent to dynamically detect the situations where it is relevant to switch from an interaction protocol to another, and to perform the "protocol switch" at run-time. For that purpose, we proceeded to a sensitivity analysis of the interaction protocols according to the context application. From the results of sensitivity analysis, we built rules which can be injected into the agents' knowledge base, allowing them to reason about the relevance of the interaction protocols.

Thesis Outline

This thesis is structured into two main parts as depicted afterward.
The first part exposes the conceptual context for the thesis throughout first chapter, and a literature review of the main MultiAgent Systems in the last chapter.

❶ Chapter 1:

Chapter 1 introduces the necessary background regarding the basic concepts of application and more precisely the Aging problem.
It also provides the background knowledge to support our work, which addresses the areas of Ambient intelligence, Multiagent Systems and Ontologies. We present the notion of interaction protocol, which is central in out work, by giving a brief overview of the different type of interaction protocols in Multiagent Systems.

❷ Chapter 2:

Chapter 2 presents in the first section a state of the art of solution methods treated in literature with Multiagent Systems for ambient assistive applications.
We identified evaluation criteria for the classification of the works, and then compared them according to these criteria. Then, we present the conclusions of this study and identify some shortcomings that will lead to our proposal.
The next section presents the problem of protocols ad-hoc implementation for the development of interactive agents. It studies state of the art for the representation, selection and development of protocols.
It describe what Adaptive multi-agent systems (AMAS) is and how they can be used for our context problem of ambient assistance. Then, we identified the shortcomings and compares several works representative of the state of the art.

The second part of this thesis is composed of three chapters and it is dedicated to detail our contributions. This part focuses on presenting our newly developed approaches solving the adaptability of cooperative Multiagent Systems for ambient assistive applications.

❺ Chapter 3:

Chapter 3 presents a usage scenario extracted from the Coalaa [N.Abchiche-mimouni *et al.* 2016] project in order to present the questions to which the thesis attempted to propose answers.

❻ Chapter 4:

In chapter 4, we develop our GAAMAAA architecture (Generating Automatically an Adaptive Multiagent systems for Ambient Assistive Applications).
Then, we present an ontology for ambient assistance which is designed for modelling the knowledge useful for the representation of the environment of assistive applications. Then, we present an ontology dedicated to the modelling of interaction protocol.
In order to demonstrate the performance of the proposed GAMAAA, we applied it on a case study for robot localization.

❼ Chapter 5:

In chapter 5, we have performed a sensitivity analysis of the interaction protocols according to GAAMAAA platform presented in chapter 4. From the results of sensitivity analysis applied to scenario described in chapter 3, we built rules which can be injected in the agents, allowing them to reason about the relevance of the interaction protocols.

Publications

✍ Amina MADDOURI, Nadia ABCHICHE-MIMOUNI, Samir OTMANE, and Jalel AKAICHI, *"GAAMAAA: Generating Automatically an Adaptive Multiagent system for Ambient Assistive Applications"*, 18*th* INTERNATIONAL CONFERENCE ON INTELLIGENT SOFTWARE METHODOLOGIES,TOOLS, AND TECHNIQUES. SoMet Malaysia 23˘25 September 2019.

✍ Amina MADDOURI, Nadia ABCHICHE-MIMOUNI, Etienne COLLE, and Jalel AKAICHI, *"Approche adaptative pour la génération automatique d'agents coopératifs, Application à l'assistance à la personne"*, 10*nd* Recherches pluridisciplinaires pour l'autonomie des personnes en situation de handicap, Handicap Paris $13-15$ Juin 2018.

✍ Amina MADDOURI, Nadia ABCHICHE-MIMOUNI, Samir OTMANE, and Jalel AKAICHI, *"Sensitivity Analysis of multiagent interaction protocols. Case-study: Robot localization in a cooperative environment"*, ACM Transactions on Autonomous and Adaptive Systems (TAAS). Being submitted.

Part 1

Context, background, and state of the art

PART 1 presents the conceptual aspects of this Thesis. It provides the necessary background regarding the basic concepts of application and more precisely the ambient assistive domain.

Besides, it introduces the theories of MAS and ontologies as main techniques adopted in this dissertation.

A literature review of the main MAS for ambient assistance application solution approaches proposed in the last decades to handle this problem is provided.

Besides, the framework of adaptative engineering MAS, are detailed.

CHAPTER 1

Context and background

1.1 Introduction

The evolution of the information and communication technologies led to the emergence of the notion of ambient environment (AE)made up of sensors, actuators, and even networked robots. The primary target of this environment is to assist, secure and control aged or disabled people in their community. Over the last decade, MAS have proven to be useful in health applications such as the ones related to ambient assisted living of Parkinson patients [I.Garcia-Magarino & G.Palacios-Navarro 2016] also other works such [Gams.Matjaz *et al.*], [Shakshuki & Reid 2015] and the famous project RobotCARE [Cesta *et al.* 2003].
Ambient intelligence (AmI) based systems aim to improve quality of life, offering more efficient and easy ways to use services and communication tools to interact with other people, systems and environments.
The design of complex systems, such as MAS, should consider models that are clear to communicate, provide support during programming, and allow the reuse and reasoning over the specification [A.Freitas *et al.* 2017a]. Therefore, we propose and investigate the use of ontologies to achieve such goals due to the fact that they can be used as a repository for these involvements and help in organising the concepts involved in the modelling, development, and verification of MAS [Artur.Freitas *et al.* 2019]. Since ontology will be playing the role of meta-model for MAS, this section briefly explains the main topics in the areas of AmI,,MAS, protocol interaction and ontology, and MAS that have led us to present this thesis.

1.2 Aging problem

Dependence is a permanent situation in which a person needs important assistance from others in order to perform basic daily life activities such as essential mobility, object and people recognition, and domestic tasks.
There is an ever growing need to supply constant care and support to the disabled and elderly, and the drive to find more effective ways of providing such care has become a major challenge for the scientific community.
According to the Department of Economic and Social Affairs of the United Nations [United Nations & Social Affairs 2019], the global population aged 65 years numbered 703 million in 2019. This number is projected to double to 1.5 billion in 2050. Globally, the share of the population aged 65 years or over increased from 6 per cent in 1990 to 9 per cent in 2019. That proportion is projected to rise further to 16 per cent in 2050, when it is expected that one in six people worldwide will be aged 65 years or over.
Globally, the number of persons aged 80 years or older nearly tripled between 1990 and 2019, growing from 54 million to 143 million; it is projected to triple again between 2019 and 2050 to reach 426 million. Between 1990 and 2019, the number of persons aged 80 or over doubled in all regions except Europe and Northern America and tripled in four of eight regions. Between 2019 and 2050, the number of persons aged 80 years or over is projected to show the largest percentage increases in Eastern and South-Eastern Asia and in Northern Africa and Western Asia.

Not only has the absolute number of older persons increased globally, but also the share of older persons in the total population has increased and is projected to continue to grow in all regions. The percentage of persons aged 65 or over worldwide has grown from 6 per cent in 1990 to 9 per cent in 2019 and is projected to increase further to 16 per cent in 2050 1.1. This will result in

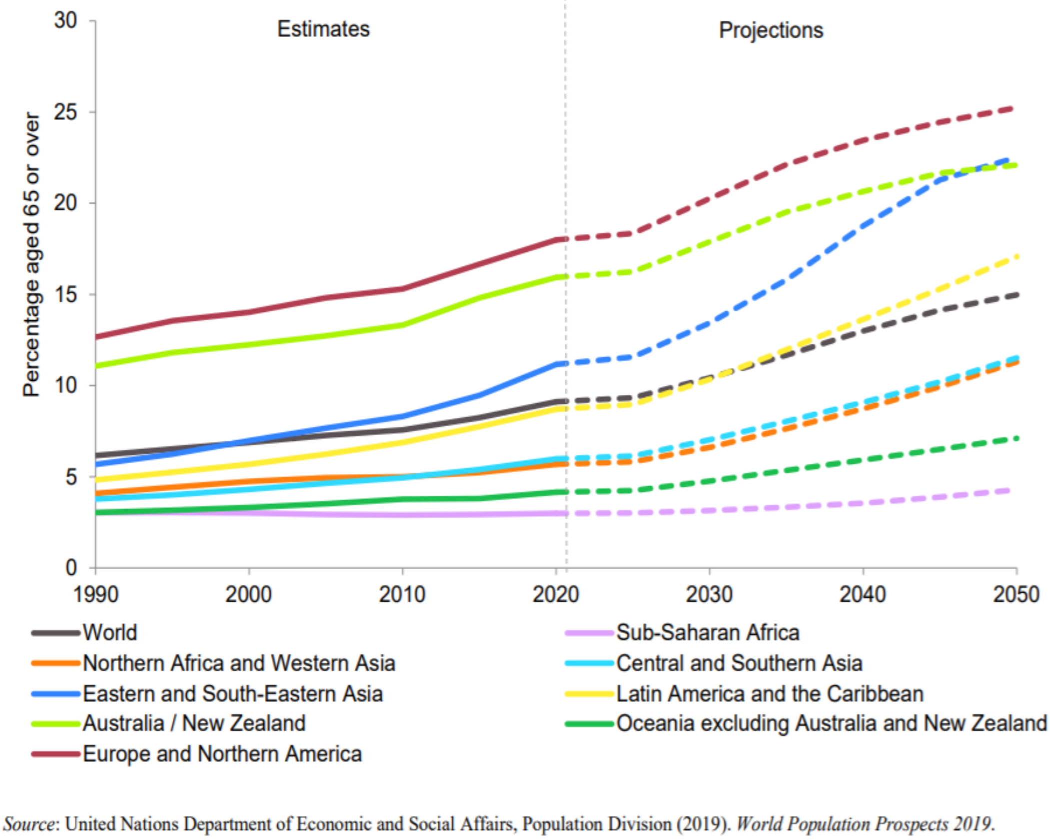

Figure 1.1 – Share of total population aged 65 years or over, by region, 1990-2050
`https://population.un.org/wpp/`

either higher costs for the services to the persons or a decrease in the quality of services or even a scenario of both higher costs and lower quality service. Assisted living solutions for elderly people using ambient intelligence technology can help to cope with this complication through providing a proactive approach that aims to address the challenges before they accumulate along with situation aware assistance.

This approach should also be capable of sustaining the autonomy of the elderly, be helpful in limiting the increasing costs while concurrently providing increasing the quality of life to the affected people. The goal is to enable elderly people to live longer in their preferred environment, to enhance the quality of their lives and to reduce costs for society and public health systems.

Today's commercially available products for emergency monitoring already use a broad range of modern technology (e.g., necklaces with emergency buttons, fall sensors in mobile phones

with wireless notification of emergency services, vital data monitoring plasters, etc.). However, they are mostly closed, stand-alone systems with a limited ability to describe the actual situation, often just too difficult for the elderly people to operate and useless in emergencies.
Moreover, as we will see the proposed solution are standardized and lack adaptation to each single individuality.

1.3 Ambient intelligence

Ambient intelligence is an emerging discipline that brings AI to our everyday environments and makes these environments sensitive to us [J.Cook *et al.* 2009b]. AmI systems can also involve AI agents and perform as autonomous systems [Gams *et al.* 2019].
Ambient intelligence research builds upon advances in sensors and sensor networks, pervasive computing, and artificial intelligence. As a reminder, the AI field was developed in 1950s as "the science and engineering of making intelligent machines", as stated by McCarthy, one of the pioneers of the field [J.Cook *et al.* 2009a]. This rather ambitious project was somewhat toned down during the 1970s when the field was the subject of several setbacks leading to an "AI winter", whose effects can still be felt today. The reason for this setback was that researchers had been too optimistic in their expectations of the breakthroughs which would be produced by the field, and did not properly take into account the inherent complexity of some of the tasks they were proposing to handle (e.g. natural language processing). This disgrace period of the AI field ended with the success of expert systems in the 1980s. These systems aim to emulate the ability of a human being to take decisions based on expert knowledge, using inference mechanisms (via an inference engine) and a rules database. However, even expert systems cannot avoid the complexity of modeling knowledge, and are still ultimately limited by the growth of their rules database. This concern, among others (such as privacy of informations) led to a new field of AI named Distributed Artificial Intelligence (DAI) [J.Cook *et al.* 2009a], where several expert systems collaborate to provide a collective solution to a specific problem.
Machine learning is a field of artificial intelligence that aims to provide a system that has the ability to learn from the data, extract important features from this data, and use these features to perform predictions or classifications on new instances of the data. This learning process starts by introducing the data to the algorithm, the algorithm then looks for patterns in the data that can be used to base future predictions on, the principal objective is to allow computers to learn automatically from the data without the need for human intervention or assistance and to allow the algorithm to adjust its actions accordingly [Tom.Mitchell *et al.* 1986]. An artificial neural network(ANN) has its roots in artificial intelligence, it is a succession of algorithms that aims to detect and recognize underlying relationships within a set of data. This system operates through a system of neurons, hence its name neural network.
Neural networks had more and more hidden layers and the high number of layers was becoming a source of problems. Indeed, from a number of layers, the neural network was no longer able to assimilate information and learn correctly.
Since these contributing fields have experienced tremendous growth in the last few years, AmI research has strengthened and expanded. AmI research is maturing, resulting in technologies

that promise to revolutionize daily human life by making the surroundings of people flexible and adaptive. Due to the high potential of emergencies, a sound emergency assistance is required for example in certain cases to assist elderly people. Comprehensive ambient assisted living solutions sets high demands on the overall system quality and consequently on software and system engineering. Therefore making user acceptance and support by various user interfaces an absolute necessity. Living assistance systems focusing on the support of people

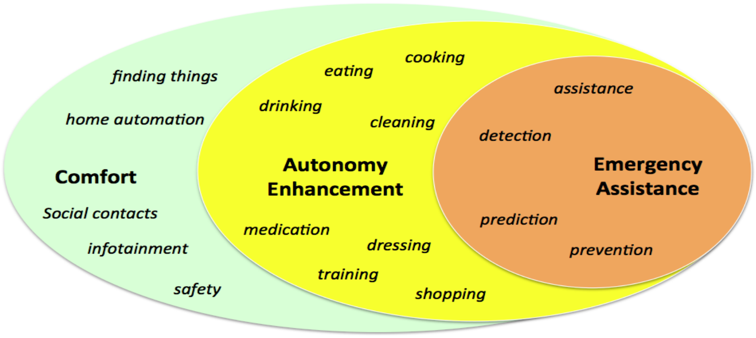

Figure 1.2 – Home care system domain

with special needs (elderly, disabled) in their own homes are called Home Care Systems (HCS) [Kleinberger *et al.* 2007]. The aim of a HCS is to allow people in need to live longer in their preferred environment at home, while retaining their independence; this is essential in cases involving handicaps or people with medical diseases. As illustrated in A.4 the HCS domain can coarsely be structured into emergency assistance services, autonomy enhancement services, and comfort services.
Finally, it is worth mentioning the survey by Preuveneers and Novais [D.Preuveneers & P.Novais 2012] on software engineering practices for the development of applications in AmI.
In this survey the authors include relevant works on the representation of knowledge through ontologies in MAS along with their reasoning capabilities. They identify the adoption of these multiagent systems as a compelling alternative for developing middleware solutions that are necessary in AmI applications.

1.4 Multiagent systems

MultiAgent Systems is a relatively recent field which can be seen as the intersection of Artificial Intelligence.
The popularity of MAS has risen significantly with the continuous increase in Internet and Web computing, which present an environment in which agents can exist and interact with one another. Before talking about MAS, we must explain the notion of an agent.

1.4.1 Agent definition and characteristics

Several definitions of an agent have been proposed. We keep here the (mostly) consensual notion proposed by Wooldridge in [Georgeff *et al.* 1999]:

"An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives".
According to [M.InvernoDavid *et al.* 1997] agents are autonomous, persistent (software) components that perceive, reason, communicate and act in someone's favour, influencing its environment.
According to Ferber [Ferber 1999] *"An agent a real or abstract entity that is able to act on itself and its environment; which has a partial representation of its environment; which can, in a multi-agent universe communicate with others agents, and whose behavior is the result of its observations, its knowledge and its interactions with the other agents."*
It follows from this definition key properties like:

- The autonomy: agents are capable of independent decision making. They operate without the intervention of human or others, and have control over their own actions and internal state [M.J.Wooldridge & N.R.Jennings 1995]. The degree of autonomy of each agent is evaluated on the basis of its proactivity and the role it has in the organizational structure. Assessment of the role of the agent in the organizational structure considers for each agent: its position in the MAS structure . The ability to take on several roles (Initiator, responder); the ability to cooperate with other agents; and sharing tasks or resources (locate Robot).

- The reactivity: Agents perceive their environment, and respond accordingly to changes that occur in it. They are responsive to the dynamics of their surrounding world. The Reactivity is evaluated by the ability of an agent to handle the exception in the system, i.e., changes in the environment.

- The communication: the agent must be able to communicate with agents as well as with human users.

In the literature, definitions [Nicholas *et al.* 1998] also give the agent other key properties:

- Pro-activeness: Agents act in anticipation of future goals, on their own initiative. They do not simply act in response to their environment, but are able to exhibit goal directed behaviour by taking the initiative.

- Socialness: Agents are able to interact with humans or other agents in order to achieve their goals. Agents can perceive their environment through sensors and act through effectors. The socialness is evaluate by:
  - Existence of communication protocol: it is important to capture the conversational messages and identify the conversational protocols that are used in communication [FIPA 2007].
  - Clarity and completeness of the communication protocol: a protocol is an orderly set of messages that together defines the admissible patterns of a particular type of interaction between entities.The clarity of the protocol is very important, because

ambiguity causes difficulty or a misunderstanding of the meaning of the message [Y.Shoham 1993].

These two additional properties allow the distinction between the two types of agents: reactive agents and cognitive agents are detailed in section 1.4.3.

1.4.2 Multiagent system definition

In the general case, the agents are not isolated but they are grouped together in a system called multiagent system (MAS). Agents are the building blocks of a MAS in which they interact in an organized manner for the purpose of achieving a common goal.
"A MAS is a distributed system, composed of a set of agents interacting, most often, according to modes of cooperation, competition or coexistence." [draa Brahim *et al.* 2001]
It emerges from this definition an essential and founding element of MAS, that of interaction protocol, detailed in chapter 3. According to [draa Brahim *et al.* 2001] the interaction can be ensured either by a transfer of information, or via the environment according to two mechanisms: perception and communication.
<u>Multiagent system examples</u>
Multiagent systems are becoming more popular and they are being used in more and more applications [Wooldridge 2009] like: coordinated defense systems, applications that include transportation, logistics, graphics, network technologies and many others. There are ongoing researches connected with multiagent systems: cooperation and coordination, organization, communication, negotiation, distributed program solving, agent learning, scientific communities, dependability and fault-tolerance. We enumerate some of the most commonly MAS:

- BitTorrent [J.M.Vidal 2008] is one of the most popular P2P based system, which is actually a MAS. Its purpose is to distribute large files like TV shows, movies, books, software. Agents in that system are autonomous and they decide what part of files will be downloaded or uploaded, from where and with what priority.

- Distributed Vehicle Monitoring Tested [Wooldridge 2009]. It is a sensor network, where an agent is responsible for a sensor. Their task is to cooperate an object (vehicle) and its recognition parameters. It is also used for predicting object further movement from one region to another with time anticipation.

- MAXIMS [Wooldridge 2009] is an email assistant system. Its task is to prioritize, forward, delete, sort and archive emails on behalf of a user. It learns from the previous action made by user and, with proper configuration, saves time of the user. Autonomous actions are based on a “confidence level”, gained by checking results of proposed action with a user.

1.4.3 Agents architectures

1.4.3.1 Reactive agents

Reactive agents represent a special category of agents which do not possess internal, symbolic models of their environments; instead they act/respond in a stimulus-response manner to the present state of the environment in which they are embedded [S.Nwana 1996]. Reactive agents work dates back to research such as Brooks [Rodnay.A.Brooks 1986] and Agre and Chapman [Philip.E.Agree & Chapman 1987], but many theories, architectures and languages for these sorts of agents have been developed since.
Guided by research such as Brooks [Rodnay.A.Brooks 1986], Demazeau [Hallenborg *et al.* 2007], Wooldridge [Wooldridge 2001] and [M.J.Wooldridge & N.R.Jennings 1995], the need for error-tolerance and fast reaction in dynamic environments have resulted in reactive architectures.
Reactive architectures do not include an internal symbolic world model of their environment and do not use complex symbolic reasoning processes. They gain their intelligence from interactions with their environment. They have task specific modules that initiate direct reactions in response to specific situations that occur in the environment.
Even if one module fails, the remaining modules can perform the tasks that they are meant to do. Thus, these architectures increase the fault-tolerance and robustness of an agent. Reactive agents cannot use their internal knowledge base to dynamically generate and pursue new goals. Instead, their goal-orientation results implicitly from the interactions with the environment. So, it is not clear whether reactive architectures are capable of having long-term goals and demonstrating goal-oriented behavior.
An example of reactive agents is that of the MANTA system Modeling an Anthill Activity due to Drogoul [Drogoul 1993] to simulate a community of ants. "In this system, the architecture of an agent includes operators of perception, selection and activation who manipulate a set of tasks." [Boissier 2001]. The Figure 1.3 illustrates an example of reactive architecture and shows that each of the percept situation is mapped into an action which specifically responses to the percept situation.

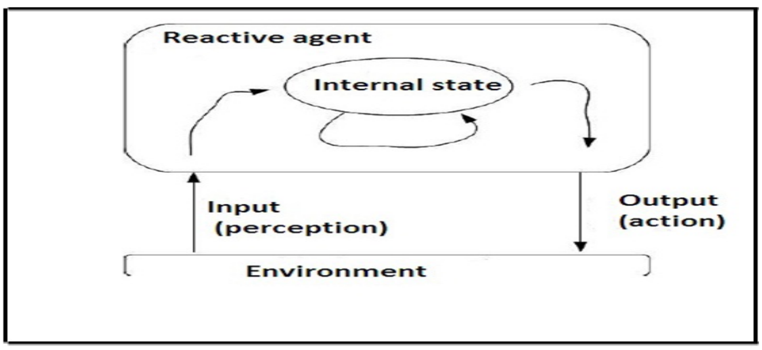

Figure 1.3 – Reactive architecture

1.4.3.2 Cognitive agents

Unlike reactive agents, cognitive agents have a symbolic and partial representation of their environment and other agents. Predominantly, these agents have the knowledge and goals. Unlike reactive agents, this type of agent has the capacity to reason on representations of the world, to memorize situations, to analyze them, to foresee possible reactions to any action, to draw from them behavior for future events and therefore to plan your own behavior [J.Ferber 1995]. An emblematic approach is that initiated in the 1980s [E.Michael.Bratman *et al.* 1988] which proposes an architecture based on three attitudes: Belief, Desire, Intention (BDI). It has since been the subject of numerous works
[Kumar & Shapiro 1994] [A.Rao & M.Georgeff 1995] [P.Busetta & K.Ramamohanarao 1998] . The Belief concept corresponds to the representation of the internal state of the agent. It is updated at all times according to the state of the environment. Desire, corresponds to the objectives set by the agent's internal state. Intent, corresponds to the goals being completed. The advantages of BDI architecture are that the design of the architecture is clear and intuitive. The functional decomposition of the agent subsystem is clear and the BDI logic has formal logic properties that can be studied. However there are many other architectures

1.4.3.3 Hybrid agent

A hybrid agent is a mixture of reactive and deliberative, that follows its own plans, but also sometimes directly reacts to external events without deliberation. This new class of agent was supported by this work [Chaib-draa & Levesque 1994]. Some authors [B.Chaib-Draa 1996]
[B.Chaib-Draa *et al.* 2001] [K.Fischer *et al.* 1995b] have been led to propose hybrid agent architectures for improve in particular decision times and action times. The hybrid agent is then conceived as combining reactive behavior and cognitive behavior. The InteRRaP architecture (Integration of Reactive Behavior And Rational Planning) [K.Fischer *et al.* 1995b] which draws on BDI concepts is an example of this type of agent. She has been used in applications such as road transport [K.Fischer *et al.* 1995b] [K.Fischer *et al.* 1995a] and robotics [M.d'Inverno *et al.* 1998].

1.4.4 Multiagent system components

According to Demazeau [Yves 1995] the basic components of MAS are the agents, the environments, the interactions, and the organisations.

With this view as shown in the Figure 1.4 the agent architectures, range from simple fine-grained automata to complex coarse grained knowledge-based systems . The environments are domain dependent. Interaction structures and languages range from physics-based interactions to speech acts [Yves 1995]. The organization constitutes a regulated space in which interactions can take place and where each member of the organization plays a role.
Ferber [J.Ferber 1995] defines an organization as a set of existing relationships between a set of agents. Among these relationships, he identifies the delegation of tasks, the transfer of knowledge, the synchronization of actions. These relationships are essential within an organization.

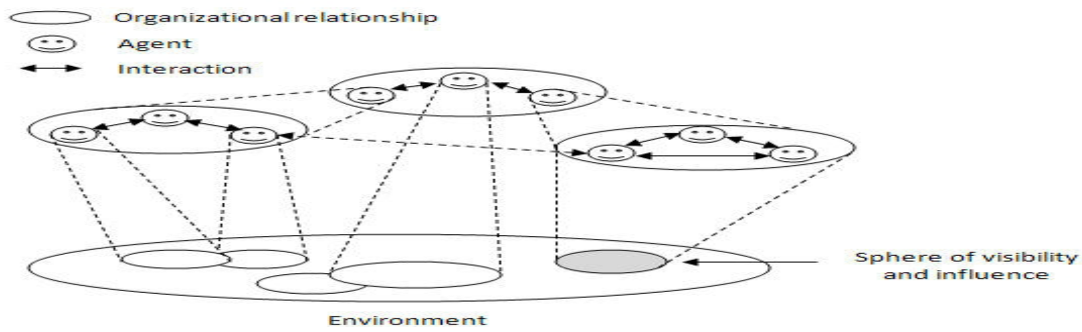

Figure 1.4 – View on a multiagent system
[Giorgini *et al.* 2006]

They constitute the basis for the sharing of tasks and knowledge, and a support for the distribution of resources and the coordination of actions. Those organizations can communicate with each other to achieve a global goal [M.Kolp *et al.* 2006].
There is one more conclusion from the figure above. In MAS a problem to solve is decomposed into multiple interacting autonomous components [Foster *et al.* 2004]. Each of them has a set of objectives to achieve and is able to perform some actions. Each component is constructed from agents, organizations and interaction patterns. Finally the complicated structures are used to describe and manage a set of relationships that emerges from agent interactions.

1.4.5 Multiagent systems software qualities

Multiagent systems have several qualities [M.Kolp *et al.* 2006] :

- Predictability: Agent should have the ability to predict and plan actions while observing and reacting to the changes in the environment. But generally it is difficult to predict the behavior of a component, because agents are autonomous.

- Security: This attribute goes with protocols and strategies for verifying if a communicating agent is the one, who has the right to do that. Also individual agents might be designated to check systems security features.

- Adaptability. Agents are (from definition) reactive for change in the environment in which they live: it also contains the level of component how do groups of agents react for changes together in the background of the whole system.

- Coordinability: Without a certain type of coordination and organization agent architecture is a low profitable one. Coordination here is connected with resources allocation, knowledge distribution, agent responsibilities. To achieve a goal agents have to cooperate, because a single agent does not possess the knowledge to solve the problem on its own. Cooperativity is described as observation of other agents' behavior and reaction on it.

- Competivity: It is mainly appearing in system, where negotiation is required. Sometimes both agents are able to benefit from negotiation, but sometimes one has to lose to let other win.
- Availability: The services provided by a MAS have to be available for usage. This is measured by agents, who are responsible for service execution coordination. Their failure decreases availability.
- Fault-tolerance. A failure of one agent does not crash the system. If MAS is really good designed, it is able to switch some responsibilities of missing agents with others.
- Modularity: It increases the efficiency of task execution, reduces communication and improves flexibility and security, but also adds more constraints.

1.4.6 Multiagent systems self-capabilities properties

While it is not true for all MAS, some interesting properties can be achieved when taking advantage of the autonomy of the agents.
This autonomy, coupled with an adequate behavior of the agents, can lead to systems able to adjust, organize, react to changes etc. without the need for an external authority to guide them. These properties are regrouped under the term self- capabilities (self-tuning, self-organizing, self-healing...). Not all MAS necessarily present all of these self-capabilities but, as a result of building a system from autonomous and locally situated agents, many MAS will exhibit them to some degree. Consequently, MAS are often relevant for dynamically taking into account changes in their environment. For example, a MAS in charge of regulating the traffic of packets in a computer network could be able to react efficiently to the disappearance of some of the relay nodes.
Agents are reactive systems that can independently determine how to best achieve their goals and perform their tasks [F.Hübner *et al.* 2007] while demonstrating properties such as autonomy, reactivity, proactiveness, and social ability. We also began to introduce concepts from the area of ontology, which is now addressed in the next section.

1.5 Interaction protocols in multiagent systems

Communications in MAS, such as in humans, are the basis of agent interactions and organization. We essentially distinguish between two modes of communication: the indirect communication which is a communication by signals via the environment, and the direct communication which proceeds to an exchange of messages between the agents. The inter-agents interaction groups and combines several types of messages. This combination results in conversational sequences which can be modeled by two approaches:

- In the first approach, the agent builds his own communication plan when he needs it to perform a task. Thus, the interaction is not defined a priory, it is emergent. The knowledge and the goals of agents govern the interaction by specifying the message to be sent

(Act of communication, recipient agents, message content,...). This approach gives the agents more flexibility in their interaction.
However, in this approach, the agent must have sufficient knowledge of the semantics of the messages and that he must be equipped with a reasoning mode allowing him to conduct its interactions.

- In the second approach, the interaction follows a sequence of messages which are specified in advance. These rules govern the exchange of messages form the interaction protocol. In addition, the interaction protocol defines the type of messages that must be exchanged. The interactive agent must comply with the rules of conversation dictated in the protocol. Compared to the first approach, interaction based on interaction protocols does not require a complex architecture within the agent. "Of course a straightforward way of reducing some of the search space of possible responses to agent messages is by using conversation policies, or interaction protocols" [M.Greaves & J.Bradshaw 1999]. Thus, an interaction protocol specifies a limited set of possible responses for a specific type of message. This interaction approach, based on interaction protocols, is the one considered in the rest of this thesis.

Agents are situated in an environment where they can perceive and modify it, and they should be able to exchange information, cooperate and coordinate their activities. The basic dimensions at Multiagent levels are the agents, the environments, the interactions, and the organizations [Yves 1995]. These dimensions are more or less addressed by Multiagent platforms (e.g., Jade, Jason).
The aim of this section is to present the notion of interaction protocol and the problem of its ad-hoc implementation for the development of interactive agents.
We begin by giving a brief overview of the theory of language acts which is at the origin of multiagent communication languages. Therefore we gives the specification of the main interaction protocols, and more particularly those defined by FIPA (Foundation for Intelligent Physical Agents). We consecrate the last section to the classification of interaction protocols.

1.5.1 The theory of speech acts

The theory of the philosophy of language was introduced in the 1960s by the work of Austin [J.L.Austin 1962]. In his book "How to do things with words", Austin shows that any utterance, allowing the accomplishment of an act which, as such, aims to accomplish something. This act is called the speech act. Austin defined three types of speech acts for an utterance:

- The locutionary act which is the act of saying something. It is satisfied when the statement is correctly formulated;
- The illocutionary act which represents the action performed by saying something, for example a request or statement;

- The perlocutionary act which corresponds to the effect obtained by saying something, for example dissuade, convince others.

Austin's work has been echoed by several researchers, in particular Searle [J.R.Searle 1969], who considered that every speech act carries an illocutionary act that falls into one of the followings five categories:

- Assertive act: the speaker expresses how the objects to which he refers are in the world. These are assertions, information, testimonies, denials, etc.
- Commissive act: the perpetrator undertakes to perform an action. These are promises, wishes, threats, etc.
- Directive act: the speaker ensures that the speaker performs an action. These are requests, questions, orders, advice, etc.
- Expressive act: the speaker manifests his mental state in the face of a state of affairs. These are excuses, thanks, congratulations, recriminations, etc.
- Declarative act: the speaker performs at the time of the utterance the action he says he is doing. These are definitions, condemnations, ratification, etc.

These works are the basis of the multiagent communication languages (i.e. KQML and FIPA-ACL) that will be presented in the following section.

1.5.2 Communication languages

The interest of communication languages is to facilitate the exchange and interpretation of messages and interpretability between agents. These languages essentially focus on how to exhaustively describe acts of communication from a syntactic and semantic point of view.
The first language that was introduced is KQML, for Knowledge Query and Manipulation Language. Originally, KQML was developed to exchange information and knowledge between knowledge-based systems. It was then taken up in [T.Finin *et al.* 1994] to describe the messages exchanged between the agents. The second language, known as FIPA-ACL[1] [FIPA 2002](FIPA Agent Communication Language), is offered as part of standardization work carried out within the FIPA organization. FIPA-ACL is an extension of the KQML language. FIPA-ACL is based on twenty-one communicative acts, expressed by performatives, which can be grouped according to their functionalities as follows:

- Passage of information: *Inform, Inform-if, Inform-ref, Confirm, Disconfirm,*
- Information requisition: *Query-if, Query-ref, Subscribe,*
- Negotiation: *Accept-proposal, Cfp, Propose, Reject-proposal,*
- Distribution of tasks (or execution of an action): *Request, Request-when, Requestwhenever, Agree, Cancel, Refuse,*

[1] www.fipa.org

Element	**Meaning**
performative	the type of communicative act
sender	the sender of the message
receiver	the recipient of the message
reply − to	the participant in the act of communication
content	the content of the message (the information conveyed by the performative)
language	the language in which the content is represented
encoding	describes the encoding mode of the message content
ontology	the name of the ontology used to give meaning to terms used in content
protocol	the name of the interaction protocol
conversation − id	the conversation id
reply − with	an identifier of the message, for future reference
in − reply − to	it references the message to which the agent is replying (specified by the reply-with attribute in the sender's previous message)
reply − by	A delay to reply to the message

Table 1.1 – The elements of an FIPA-ACL message

- Error handling: *Failure, Not-understood*

A FIPA-ACL message can contain some or all of the elements described in table 1.1. The elements required to convey a message change depending on the situation. If an agent does not recognize or cannot process one or more items, then they can respond with the Not-understood message. Here is an example of a FIPA-ACL message that is sent by Agent A to Agent B:
(inform
:sender A
:receiver B
:reply-with devis12
:language Prolog
:ontology Ordinateur
:content prix(HP,1500 EUR))
:conversation-id conv01
:reply-by 10 min)
Agent A informs his interlocutor that the price of an HP computer is 1500 euros. The content of the message is expressed in the Prolog language. The ontology used is that of computers. This message is part of a conversation with the identifier conv01. Agent B is constrained by 10 minutes to follow up on this message.
Communication languages are fully used for the specification of interaction protocols. Interaction protocols are used by agents to govern their interactions.
In the following section, we give some definitions, existing in the literature, of interaction protocols.

1.5.3 Definition of interaction protocol

Several definitions of the concept of interaction protocol are found in the literature. We adopt the definition of Cossentino [A.Chella *et al.* 2004] who describes an interaction protocol as follows: ”An agent interaction protocol is a common pattern of conversations used to perform some useful tasks: the protocol is often used to facilitate simplification of computational machinery needed to support a given dialogue task between two agents”. The second part of this definition sums up the purpose of the interaction protocols from a functional point of view. The interaction protocols differ according to whether the agents are competing or sharing common goals.
In the Gaia methodology [F.Zambonelli *et al.* 2003], the authors used the following definition: "a protocol can be viewed as an institutionalized pattern of interactions. That is, a pattern of interactions that has been formally defined and abstracted away from any particular sequence of execution steps, to focus attention on the essential nature and purpose of the interaction, rather than on the precise ordering of particular message exchanges."
Greaves and his colleagues give in [M.Greaves *et al.* 2000] the following definition: "Interaction protocols are descriptions of standard patterns of interaction between two or more agents. They constrain the possible sequences of messages that can be sent amongst a set of agents to form a conversation of a particular type."
From these definitions come four fundamental notions that characterize an interaction protocol:

- An interaction protocol is an interaction pattern. This explains the need to represent it in a generic way, regardless of the application context.
- Each interaction protocol has a purpose.
- An interaction protocol involves two or more agents. Each of these agents plays a role in identifying it during the interaction.
- The interaction protocol defines the rules that govern an interaction. These rules define the ordering of messages, as well as the actions to which the protocol appealed.

1.5.4 Classification of interaction protocol

Mazouzi classifies in [H.Mazouzi *et al.* 2002] the protocols into four main classes: coordination protocols, cooperation protocols, negotiation protocols and auction protocols.

- Coordination protocol Coordination protocol in resource-constrained settings, the coordination results in individual behavior aimed at serving one’s interests while trying to satisfy the overall goal of the system. A well-known coordination protocol is the contractual network (contract-net)[R.Smith 1980]. In this section, we describe some interaction protocols that will serve as a starting point for a detailed analysis of the activity of interaction protocols. Iterated Contracted Net Protocol (ICNP) as depicted in Figure 1.5 embeds in each agent a protocol role:
  - Initiator: is played by the manager agent, he sends the call for proposal (CFP), including the specification of the request to the participant agent (agent whose role is

participant).

- Responder: receives the proposal, he can refuse the *CFP* or accept it, and then sends the proposal.

The manager receives the proposal and the refuse messages, and then it evaluates the proposals, selects the best one. Then the Initiator notifies the participants its decision to the accepted and rejected responders. If all agents failed, the manager returns to send *CFP*.
The FIPA Iterated Contracted Net Protocol is an extension of the basic FIPA Contract Net IP, but it differs by allowing multi-round iterative bidding.

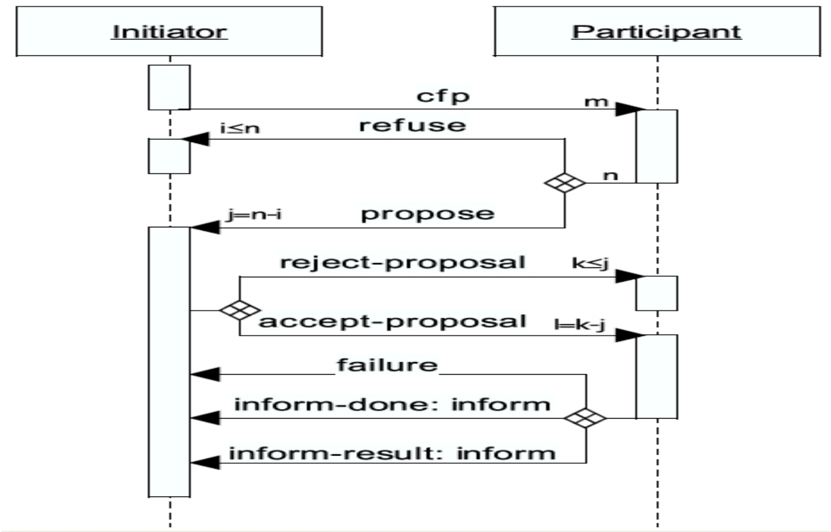

Figure 1.5 – Contract Net Interaction Protocol

- Cooperation protocol ([R.Smith 1980]), the interaction is the support of the materialization of the cooperation. The cooperation is a globally observable concept in a MAS. Meanwhile, the interaction is observed between agents. Collaboration is the manner whereby a set of agents form a cooperative entity, a group, a coalition, an organization, or a team. Coalition protocol consists in forming temporary grouped agents to achieve a goal as shown in Figure 1.6. Following the receipt of an *InitEffect*, each agent A tests his ability to solve the problem, if this capacity is sufficient, he launches directly into the resolution of the request by sending *InitCoal*.

- Negotiation protocol ([P.Mathieu & M-H.Verrons 2005]) are often used as trading protocols when the agents have different purposes. It will often be a matter of resolving the conflicts between agents. The main features of the negotiation are: the agents' belief model, the protocol followed in the principle of the negotiation and the decision procedure used by each agent to determine its positions, its concessions and its criteria for the agreement.

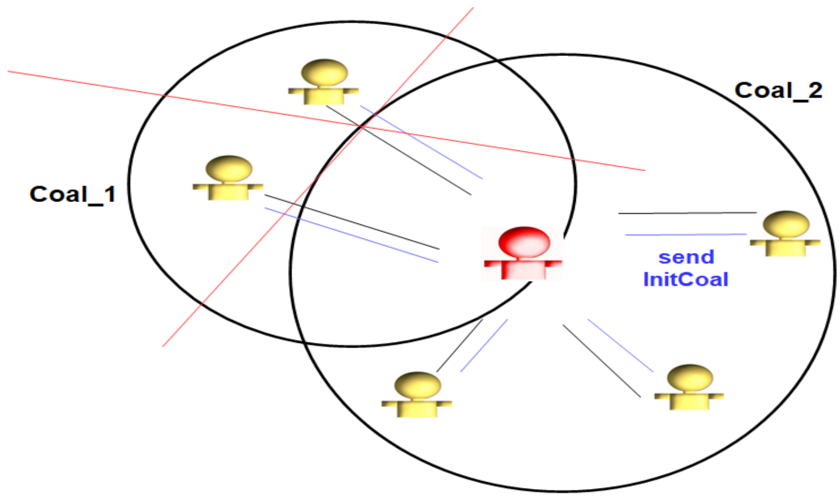

Figure 1.6 – Coalition Protocol

- Auction Protocol ([M.Reck 1994]) is based on an auctioning mechanism, widespread in e-commerce. It requires an organization and a large number of communications. Its main goal is to determine the price of an item. A methodology is predefined and several contractors who wish to purchase the item come together. The aim is that the article is either to be sold in a public way or to the highest bidder.

1.6 Ontology

1.6.1 Definitions

In the context of Artificial Intelligence, an ontology is identified with a set of formal terms representing knowledge.
Gruber [R.Gruber 1993] define an ontology is a specification of a conceptualization.
The exact meaning depends on the understanding of the terms "specification" and "conceptualization". Explicit specification of conceptualization means that an ontology is a description (like a formal specification of a program) of the concepts and relationships that can exist for an agent or a community of agents.
The history of artificial intelligence shows that knowledge is crucial for intelligent systems. To have truly intelligent systems, knowledge needs to be captured, processed, reused, and communicated. Ontology supports all these tasks.
We summarize some selected definitions of the well known ontology definitions.

Definition1: According to Neches et al. [R.Neches *et al.* 1991], ontologies are considered as a kind of *"top-level declarative abstraction hierarchies represented with sufficient information to lay down the ground rules for modeling a domain"*. However, an ontology defines the basic terms and relations, including the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary;

Definition2: According Guarino [N.Guarino 1998] an ontology is defined as an engineering artifact, constituted by a specific vocabulary used for describing certain reality, in addition to a set of explicit assumptions regarding the intended meaning of the vocabulary words.

1.6.2 Ontology components

The formalization of Knowledge in ontologies is based on four components: classes, relations, functions and axioms [M.Hadzic *et al.* 2009].

Classes (or concepts): is a collection of objects that share specific restrictions, similarities, or common properties. A property expresses relationships between concepts. An individual (instance, object, or fact) is an element of a concept. A concept can be a description of a task, action, function, strategy, reasoning process, etc. For example, all mammals share the same characteristics, except for the ability to talk.

Relations (or Properties): in ontologies, relations describe the means in which individuals (instances or particulars) are related. To define a relationship, it is necessary to define what is called an **RDF** [2] triple (domain, property,object). In other terms, relations represent a form of interaction between concepts in the same domain. For example, a sound sensor, a property Belong to may have as domain a class sound sensor and for picture a class ROOM: will connect instances of the sound sensor class to instances of the room class.

Functions: are a particular type of relations, the relations "Price-of-a-new-computer" is example of functions. For example, the function aims to calculate the price of a new computer depending on the CPU type and speed, hard-disk storage capacity and the capacity of memory storage.

Axiom: represent assertions formulated in a logical form that together comprise the core knowledge that the ontology describes in its domain of application.

The term ontology empowers the execution of semantic reasoners that provide functionalities such as consistency checking, concept satisfiability, classification, and realisation. In other words, reasoners are able to automatically infer logical consequences from a set of axioms.

- Pellet [Sirin *et al.* 2007] is one example of a semantic reasoner implementation over OWL ontologies. OWL (Web Ontology Language) is a language for processing web information and a semantic web standard formalism to explicitly represent the meaning and relationships of terms [Bechhofer *et al.* 2004].

[2] RDF: Resource Description Framework,Developed by the W3C, RDF is the basic language of Semantic Web. Is a graph model intended to describe formally Web resources and their metadata so as to allow the automatic processing of such descriptions.`https://www.w3.org/TR/2002/WD-rdf-concepts-20020829/`

- Jena is a Java framework for building Semantic Web applications. It provides a extensive Java libraries for helping developers develop code that handles RDF, RDFS , OWL and SPARQL in line with published W3C recommendations. Jena includes a rule-based inference engine to perform reasoning based on OWL and RDFS ontology.

1.7 Ontologies and multiagent systems

One of the first approaches in literature to consider ontologies to enhance an agent oriented programming language was Agent Speak-DL [Álvaro F. Moreira *et al.* 2006a]. However, Agent Speak-DL focuses on using ontologies during agent reasoning, instead of modelling aspects of MAS in ontologies. The advantages pointed out of integrating agents and ontologies are [Álvaro F. Moreira *et al.* 2006a]: (i) more expressive queries in the belief base, since results can be inferred from the ontology and thus are not limited to explicit knowledge; (ii) refined belief update given that ontological consistency of a belief addition can be checked; (iii) the search for a plan to deal with an event is more flexible because it is not limited to unification, i.e., it is possible to consider subsumption relationships between concepts; and (iv) agents can share knowledge using ontological languages such as OWL.
As far as we know, the use of ontology to support modelling, development, and verification of MAS in the context of system engeneering [A.Freitas *et al.* 2017b]. In this context, a MAS can be better designed, expressed, and communicated, and a specific modelled project can be more easily verified and converted to code. The use of ontology which represent an environment adds three important features to existing multi-agent approaches [Álvaro F. Moreira *et al.* 2006b]:(i) ontologies provide a common vocabulary to enable environment specification by agent developers (since it explicitly represents the environment and agent essential properties, defining environments in ontologies facilitates and improves the development of multi-agent simulations);(ii) an environment ontology is useful for agents acting in the environment because it provides a common vocabulary for communication within and about the environment (it allows interoperability of heterogeneous systems); and (iii) environment ontologies can be defined in ontology editors with graphical user interfaces, making easier for those unfamiliar with programming to understand and design such ontologies.

1.8 Conclusion

Through this chapter, we have been able to cover all the aspects which concern ambient assistance and agent technology. The latter as the basic entity and key concept of this paradigm had an important space in this study. We have seen that this entity has an internal structure, an environment, a type which could be reactive, cognitive or hybrid and has a life cycle which defines its individual behavior.
Interaction in a MAS is essential because it allows agents to reach their goals and accomplish their tasks. Since, we have presented the multiagent interaction protocols and the different forms it can take: communication, cooperation, coordination and negotiation. Regarding com-

munication, we have presented the communication language FIPA-ACL. We have also defined coordination and its planning role, cooperation and its principle, negotiation and auction protocol. We have deepened the technique of protocol-based interaction by showing their interest and importance in a MAS.

Ontology are involved in AI systems in order to model (represent) what we call ambient context. The later is composed of the smart object can for example have the role of a facilitator in helping a robot to locate, browse and search an object. The MAS are lead as solution to perform a service to the person by making the CO of the environment communicating and interacting. Conversely, the robot can be seen as a communicating object put by services rather than by personal assistance in loss of autonomy.

In this Chapter, we presented the aging problem and the basic notions related to MAS, interaction protocol and ontologies. We started to link these areas among each other, in this chapter. These interconnections will be explored in more details and will be described from ambient assistance point of view and MAS engineering in the next chapter.

Chapter 2

State of the art

2.1 Introduction

Ambient Assisted Living (AAL) refers to a system of auxiliary sensors that make the independent life of a person more comfortable and easier in the home environment. These auxiliary sensors enhance safety and comfort and encourage individuals to live alone. AAL systems also enhance the quality of life because they include services that link auxiliary sensor products, concepts and technologies [Mahdavi-Hezavehi *et al.* 2017].

In an ambient environment, a diversity of surveillance management troubles can be solved by means of the agent modelization procedure, such as cooperation among the different connected objects units (distributed knowledge), collection and access to information, etc.

Indeed, by introducing a cooperative MAS process, a great deal of these issues might well be resolved. Thanks to its characterizing features (autonomy, sociability, and reactivity), MAS is adapted for the particular purpose of investigating these problems. Hence, in this work, one argues that MAS stands as an adequate tool whereby the healthcare relating problems, can be addressed, while some real-life examples have been selected to demonstrate the application of such technology to deal with a diversity of the field-related concrete problems, the focus of this work is mainly on the Ambient Assisted Living (AAL).

This chapter presents in the first section a state of the art concerning the contribution of MAS to the field of personal assistance by giving some of the most known systems.

All the approaches mentioned hereafter have significant limitations concerning adaptability.

Since the environment is dynamic regarding the available CO, in addition to the diversity of situations, consequently the interaction mechanisms between agents must be dynamically and adaptively designed. Our aim is to structure and organize the interactions between the agents to obtain an adaptive MAS. The MAS are naturally adequate for the design and the control of such complex systems.

The main objective of agent-oriented software engineering is to develop methodologies and tools that facilitate the development and maintenance of multiagent applications [A.Tveit 2001]. So, one of the objectives of this chapter is to raise some design and implementation challenges.

The next section presents the problem of interaction protocols ad-hoc implementation for the development of interactive agents. It studies state of the art for the representation, selection and development of protocols. It describes the concept of an Adaptive multiagent systems (AMAS). In the section after, the shortcomings of this concept are identified and compared, and several works representative of the state of the art are demonstrated.

2.2 Multiagent systems for ambient assistive applications

Ambient intelligence tries to adapt the technology to the needs of individual and their environment based on ubiquitous computing. In this context, MAS comprise one of areas that make a strong contribution to the paradigm of adaptability in ambient assisted applications. Over the last decade, various projects have been addressed on adaptability technology at home based on MAS to improve the security and automating medical staff's work.

2.2.1 Ambient assisted applications in static environment

IDorm project

The IDorm project [D.Rombach *et al.* 2011] is one of the pioneering projects in ambient intelligence. It is designed to assess an ambient environment composed of three categories of objects: static CO associated with the building, a robot and mobile devices. IDorm architecture is based on a MAS that manages the operations of all the environmental sensors and the robot. One agent named the sensor agent controls the sensors, and controls the robot. The sensor agent receives the different measures from sensors and controls actuators, which are linked to sensors like a pan-tilt camera. The robot agent acts as a data server and coordinate exchanges of information between the user and the robot. It controls the navigation of the robot by combining different functions such as the obstacle avoidance or the search for targets. This continuous project has been upgraded to iSpace where learning capabilities are added to the system. The new version of the system learns about the behavior of the users.

While this contribution is significant because it is one of the precursors in the domain, the adaptiveness and the distributed control, inherent in MAS have been insufficiently exploited. Only two agents are present in the MAS.

ALZ-MAS project

ALZ-MAS [J.Corchado *et al.* 2008] is an Ambient assisted Application based MAS aimed at enhancing the assistance and health care for Alzheimer's patients. ALZ-MAS takes advantage of the cooperation between intelligent agents and the uses context-aware technology providing a ubiquitous, high-level interaction with flexible interfaces between users, system and the environment. System structure has five different agents that are embedded into deliberative agents based on the BDI[A.Rao & M.Georgeff 1995] architecture as can be seen in Figure 2.1. ALZ-MAS makes use of RFID, Wifi-Network and ZigBee devices, providing the agents automatic and real-time information about context. The device agent interacts with the Zigbee to control physical services.

Furthermore, the admin agent processed all information obtained. The essential aspect in ALZ-MAS is the use of a set of technologies which are integrated to agents so that to provide multiple services. Moreover, a cooperation agent is employed to plan and schedule the tasks defined for the nurses connected to the system. The strength of this approach is that it provides a middleware, which is a BDI-based MAS system, allowing it to be easily improved in terms of reasoning ability. But this system is used for the nurses and does not interact with the person at home.

RobotCARE project

The RobotCARE [Bahadori *et al.* 2006] project is a Research and Development activity running under the Ambient Assisted Living Joint Program, which is co-funded by several European countries. Its primary objective is to fall detection and the person monitoring at home by Smart camera. The system has a plurality of agents. Each agent creates many services. The agents must coordinate with each other's and have real mechanisms to ask their services. All this is made possible by the ASF system: Active Supervision Framework. ASF has two very

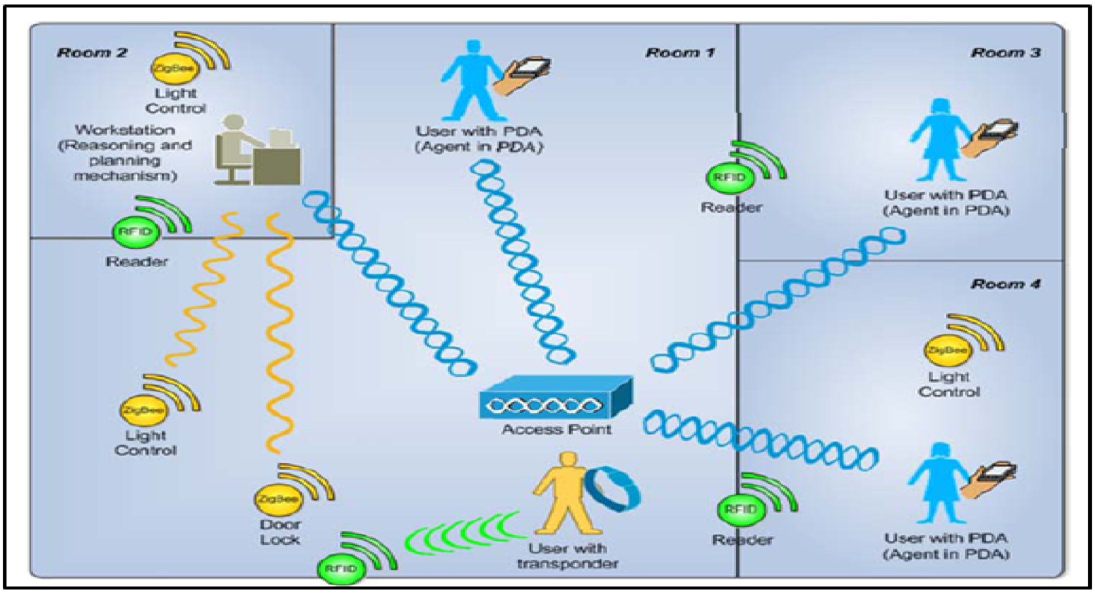

Figure 2.1 – ALZ-MAS basic schema
[J.Corchado *et al.* 2008]

important components: The agent who locates and monitor people and the agent who controls the execution. The project RobotCARE combines different techniques of artificial intelligence such as agents and computational vision. But its structure is rigid because event management is a bottleneck for requests and responses.

PalliaSys project

The PalliaSys project [Moreno 2003]is an ambient assisted application based on MAS aimed at enhancing the care given to palliative patients. These patients are in a very advanced stage of serious illnesses. One of the basic goals of the PalliaSys system is to improve the process of gathering and to collect information of the palliative patients. The work takes advantage of the cooperation between intelligent agents and the use of context-aware technologies providing a ubiquitous, high-level interaction with flexible interfaces between users, system, and environment. PalliaSys makes use of Palliative Care Unit (PCU) to store the information in the database, the access to data through sensors to provide the agents automatic and real-time information about context, and the use of PDAs by doctors when they are performing a visit (at the medical center or the patient's home). System structure has six different agents running in the multiagent systems of the PCU:

The Communication Manager interacts with the sensors and with the mobile doctor agents running on PDAs to control physical services. The database Wrapper agent controls the access to the database. Furthermore, the PCU Coordinator provides the interface between the Communication Manager and the Doctor Agents. This list could be executing in the desktop computer of each doctor to obtain the information of his patients easily. Patient Agent is responsible for continuously monitoring the evolution of one patient. Finally Data Analyzer agent may access the database and analyze this information to create patient models, using unsupervised machine learning techniques such as clustering.

CAMAP project

CAMAP is a framework presented by Ferrando and his colleague in [S.P.Ferrando & E.Onaindia 2013]. This work lies in adaptive and intelligent behavior of ambient assistance. CAMAP system aimed to apply multiagent planning based on argumentation defeasible logic for deciding the action that meets the needs of the patient. The argumentation and the partial planning methods [L.Amgoud & H.Prade 2009] are used as a mechanism for dealing with context adaptation. The overall goals of CAMAP protocol are to collaboratively and progressively refine the base plan until it becomes a solution plan. For these reasons, CAMAP seems to be relevant to find optimal solutions and could be less efficient if one must obtain a result. Indeed, such mechanisms ensure adaptive solutions without taking into account the optimality of the solutions. So, they can provide quick solutions, which can be refined if needed.

2.2.2 Ambient assisted applications in dynamic environment

COALAA Project

COALAA Project (Coalitions for Ambient Assisted living applications) as can be seen in Figure 2.2 [N.Abchiche-mimouni *et al.* 2016]: it is a MAS based on the formation of coalitions of agents tending at offering a service adapted to the person. Each COALAA agent encapsulates a connected object and defines locally and proactively the time and manner of the contribution to the service required for the person.

A more general notion that a service, called effect, has been introduced. An effect can be special lighting at a specific location of the residence or the location of a robot. The coalition-based protocol allows the MAS to configure itself to provide a solution about the availability of the connected objects and the compliance with the criteria of the urgency of the situation and the level of intrusion tolerated by the person. The critical limitation of the COALAA architecture is the consideration of an intelligent list of agents at the beginning of the implementation. Of course, the list of the created agents corresponds to the objects present in the environment during the system initialization step. However, since agents are created manually, the adaptability of COALAA is limited. This can be improved by adding an automatic authentication process in the system.

FRIENDLY KIND project

FRIENDLY KIND [F.Aielli *et al.* 2016] is a project for the assistance of the elderly. This work aims to ensure personalized monitoring of the patient assisted by RMA (Remote Monitoring Agent) agents. To achieve their goals each patient is monitored by a PMA (Patient Monitoring Agent). This agent is characterized by autonomous, responsiveness, sociability, user-friendliness, location, the intensity of knowledge, system dynamism and monitoring performance. To manage the integrability between FRIENDLY KIND components, they have proposed a domain ontology used to bring together the common vocabulary of the domain, and link the set of the knowledge source, pattern rule for the verification of the execution to ensure that the actual dynamic system is the expected one.

The FRIENDLY KIND system provides flexible access to dynamic, heterogeneous and knowledge-

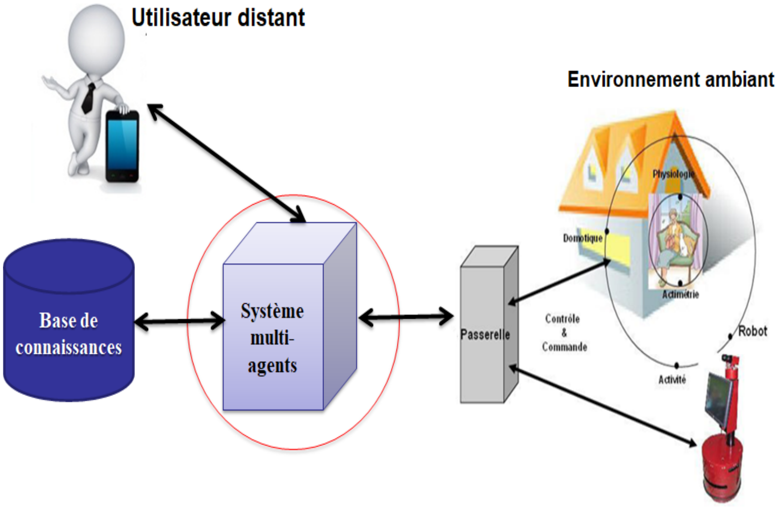

Figure 2.2 – Coalaa basic architecture
[N.Abchiche-mimouni *et al.* 2016]

based distributed sources in a dynamic environment consisting of computing entities, sensors, devices and services. The FRIENDLY KIND project uses several disciplines of artificial intelligence such as expert systems, ontologies and MAS. These techniques ensure that FRIENDLY KIND works in complex, open and dynamic computing environments including different software components, physical devices, and sensors including portable health monitoring devices, intelligent software agents. The main limitation of the used MAS is evident through its rigid structure in the design of the agents. Given the centralization of all activities at the level of a PMA agent, the reactivity time of the latter is not optimal. The system cannot respond to urgent situations or adapt to changing contexts. All this without counting that the interactions between agents are limited in number. The introduction of the communication between agents would improve the weak points of this work.

Protocol-driven agents

Other research was initially published in [D.Ancona *et al.* 2015] and was extended in [A.Ferrando *et al.* 2016] by a framework for defining protocol-driven agents. The main idea behind this work is to integrate trace expressions for protocol modeling and to insert generic events into the protocol rather than behavior based on perceived events for security-critical applications. This system aimed to apply a trace-based MAS to meet the needs of newborns who may have been suffering from hypoglycemia.

The work has the advantage of reducing the inappropriate use of hospital resources by supporting screening decisions based on the medical protocol. However, this approach has a significant limitation: a list of agents is preconceived at the beginning of the implementation and the system does not allow taking into. **SHRS**

Smart home reasoning system (SHRS) published in [Mekuria *et al.* 2019] based on a multi-agent system technologies and probabilistic logic programming technique called ProbLog.

The present paper utilized the combination of MAS and probabilistic logic programming techniques to tackle uncertainty related issues in AAL environments. Specifically, in this paper, the smart home system is modeled in terms of collaborative intelligent agents, and probabilistic reasoning is utilized to give the agents an ability to make a decision under an uncertain situation. In order to realize this architecture, four kinds of room-level agents are proposed:

- Device agents (DA) responsible for controlling a specific sensing device(s), continuously monitors the changes in the environment and combines low-level sensor readings with other data in the home to generate high-level contextual information about the state of the home or its inhabitant. On the other hand, a device agent which controls the operation of an actuator(s) put into consideration the current situation of the home and its inhabitants, while executing a user command or during a self-adaptation process.

- Service agents (SA) are general purpose agents which provide house level information, that is not specific to a single room or space in the home environment.

- Reasoner agents (RA) in the proposed MAS architecture, each room is equipped with a reasoner agent (RA), which is responsible for the automatic control of the room environment and its adaptation to the inhabitant's needs. The decision-making unit of this agent is designed based on a probabilistic logic programming technique called ProbLog. RA possesses the ability to act under uncertainty and perform well with erroneous sensor data and ambiguous user commands.

- Negotiator agents (NA) enables other agents in the system to exchange information about the state of the environment they are operating in. NA discusses the information exchange processes between the intelligent agents, based on the FIPA Request Interaction and Contract Net Protocols, thus at different times, it acts as the initiator or participant of an information exchange process, or it acts as the manager or contractor of a negotiation process over reasoning tasks.

The critical limitation of the smart home reasoning system is the application and advantage of contract net protocol for the distribution of the reasoning task are not discussed in detail. All this without counting that the interactions between agents are limited in number. The introduction of the communication between agents would improve the weak points of this work.

2.2.3 Comparison and discussion

A comparison of related work is depicted in table below 2.1, the table shows the work of personal assistance based on MAS in dynamic environment, the use of these works is to ensure the support of the person with loss of autonomy.

System	COALAA	FRANDLY AND KIND	Protocol-driven agents	SHRS
Usage	Make the robot's localization function more reliable thanks to cooperation with communicating objects to monitor the elderly	Intended for the assistance of the elderly, aims to provide personalized monitoring of the patient	Support and monitor newborn patients with hypoglycemia with MAS.	Tackle uncertainty related issues in AAL environments, specifically, the smart home system.
Technology	A MAS based on the Coalition Building Protocol.	The FRIENDLY and KIND project uses several disciplines of artificial intelligence such as expert systems, ontologies and multi-agent systems.	The trace expressions modeling protocols-driven are made generic to model any type of protocol (application to medical protocols).	Use multi-agent system technologies and probabilistic logic programming technique called ProbLog.
Negative − points	The absence of the dynamics of interactions between agents	The rigid level structure of agent design	The system does not take into account that the number of agents changes. The system cannot respond to urgent situations.	The interactions between agents are limited.

Table 2.1 – Comparing related work of ambient assisted applications in dynamic environment

All the approaches mentioned above have significant limitations concerning adaptability and self-adaptability. The central objective of the works presented here is to propose an intelligent and cooperative agent model of the interaction between the various CO. The CO environment is exploited to offer a service (fall detection, localization...) adapted to the needs of the person. As the environment is dynamic regarding the available CO or otherwise and the diversity of situations, consequently the cooperation mechanisms must be dynamically and adaptively designed. The MAS are naturally adequate for the design and the control of such complex systems. Our approach aims at structuring and organizing the interactions between the agents for the automatic generation of cooperative and adaptive agents. Therefore, our MAS emphasizes on an interaction model, an ontology-based architecture and a library of protocols which are dedicated to ambient assistive applications. The dynamic protocol-based cooperation presents various specific requirements that must be taken into consideration during all the phases of the protocol development cycle.

- The first requirement is essential for the use of the protocols. It stresses that protocols are sharable and exchangeable between heterogeneous agents operating in an open and dynamic environment.
- The second requirement deals with the richness of the expression of the protocol specification language: the protocols must be sufficiently rich to cover the different types of situations (task allocation, negotiation, auction, etc.), and to provide agents with the possibility of interpreting and reasoning about the purpose of these protocols.

- The third requirement concern is the maintainability of the protocol the case it would be necessary to modify its code.
  For the sake of simplicity, the agent will be able to use only a limited amount of interaction protocols specified in advance by the developer regarding the type of the application. However, in the case of a dynamic architecture where the agents are evolving (automatically created if needed), it may be necessary to automatically export a specific interaction protocol within the agent newly created. This is the case in a dynamic environment where sensors, cameras, actuators (the COs) could be removed or added.

- Finally, the fourth requirement results from the previous one. It reveals that agents should not be designed with their "hard-coded" roles; they can instead integrate these roles dynamically at run-time. The respect of each of these requirements eliminates the specific engineering issues related to different phases of the protocol development cycle.

A comparison of related work is depicted in table below 2.2, the table resume the work of personal assistance based on MAS by comparing the type of environment, the adaptive and the self-adaptaive of MAS.

System	IDorm	ALZ-MAS	RobotCA	PalliaSys	CAMAP	COALAA	FRANDL AND KIND	Protocol-driven agents	SHRS
Environment	Static	Static	Static	Static	Static	Dynamic	Dynamic	Dynamic	Dynamic
Adaptive MAS	Untreated	Semi-treaty	Untreated	Untreated	Semi-treaty	Treaty	Untreated	Treaty	Treaty
Self-adaptive MAS	Untreated	Untreated	Untreated	Untreated	Untreated	Untreated	Untreated	Untreated	Semi-treaty

Table 2.2 – Comparing related work of ambient assisted applications in dynamic environment

Absence of a global approach regrouping the full development cycle of MAS, from the specification of the ambient environment to the use of the MAS.

2.3 Engineering multiagent systems

The main objective of agent-oriented software engineering is to develop methodologies and tools that facilitate the development and maintenance of multiagent applications [A.Tveit 2001]. The selected works are examined from three points of view which are: (1) their representation model, (2) their execution and selection models (when they are defined) and (3) relative to the evaluation criteria considered important for these models.
We examine in detail the work that we believe to be the most representative to cover the three points of view as well as possible.

2.3.1 Protocol management system

In the context of coordination in MAS, this work [W.Bouaziz & E.Andonoff 2015] proposed an approach for the representation, selection and execution of an interaction protocol in a MAS. It consists in a Protocol Management System (PMS) which considers protocols as first-class entities to be modeled separately. The execution of the protocols is delegated to a specific component, which is the moderator. They propose a high-level protocol ontology to coordinate the distributed systems. This shared ontology is downloaded by the agents who have to integrate it into their code for the sake to play a role in the protocol. The ontology is composed of three levels [W.Bouaziz *et al.* 2009]:

- The first level is the most abstract one. It corresponds to the ontology of the protocols which defines the invariant structure shared by all the protocols.
- The second level represents the specification of a protocol (e.g., net contract) through the instantiation of this ontology.
- The last level is the most concrete one, since it consists in the execution of the conversations of the protocol. From a protocol selection and execution point of view, the PMS architecture (see Figure 2.3) supports the entire life cycle of protocols from specification to execution.

This system acts as a mediator between the agents involved in conversations. Thus, the architecture integrates agents dedicated to the specification, the selection and execution of the protocol. In the PMS, agents imply two sources of knowledge: a protocol ontology and the domain ontology. A selection model allows the agents to choose among the protocols or the available

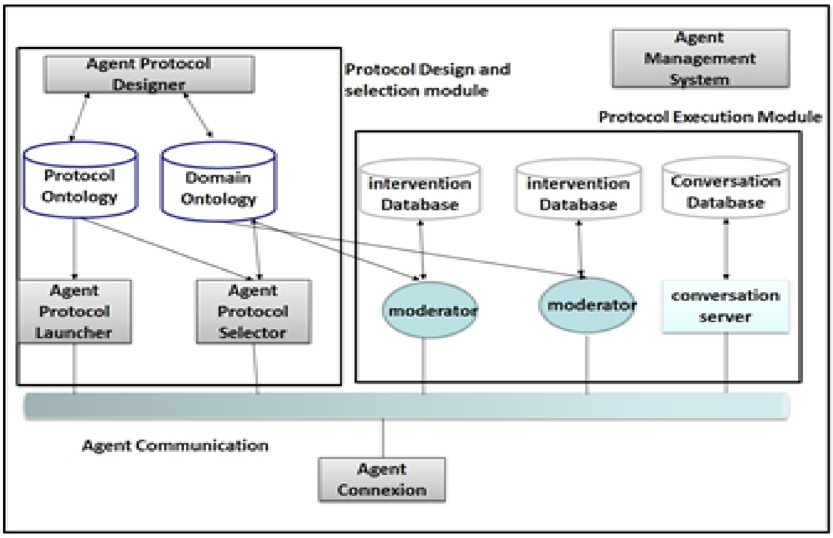

Figure 2.3 – Protocol Management System
[W.Bouaziz & E.Andonoff 2015]

conversations those which are appropriate for achieving their goals. They use the SWRL language [I.Horrocks *et al.* 2004] to specify protocol selection queries, and an algorithm to match

requests and protocol descriptors. This correspondence takes into account both the profile of the protocol and its behavior. A protocol execution model defines how the protocols are executed. Firstly, this model proposes a method of automatic generation of roles (behavior relating to each participant) from the overall description of the interaction (description of the protocol). These behaviors are later transformed into executable specifications. The transformation process is an algorithm for mapping XML specifications (describing roles) to Java skeletons (implementation role). XSLT is used to describe the mapping rules. They proposed an agent architecture that integrates a micro-engine responsible for managing the execution of roles to help agents to perform their roles dynamically.
Finally, it should be noted that the taxonomy of the ontology is specific to the coordination protocol. Regarding the dynamic execution of the protocols, the choice of the agents to participate or not in conversations is not taken into account. Another limitation is revealed the ability of the agents to operate the protocols automatically. The automatic generation of the agents is not considered. Indeed, the designer of the system has to code the agents. Moreover, the evolution of the number of agents is not taken into account.

2.3.2 Modeling, programming and verifying MAS

In the context of MAS modeling, the work [A.Freitas & R.Bordini 2017] proposes an approach for modeling, programming and verifying a MAS. The approach applies a semantic web technology for a MAS generation. It is based on the definition of an ontology as a global model which integrates the main characteristics of a MAS.
Programmers have three distinct starting points to encode a MAS, which allow them to have a single, unified and complete metamodel combining of all these dimensions. From a modeling point of view, this approach advocates that the MAS is first modeled by this higher ontology, which uses a unique formalism to encompass the global characteristics of the MAS.
The scientific contribution of this work resides mainly in the use of an ontology to address different dimensions which are: the agent, the environment and the organization of a MAS. In these terms, it can be considered as a language, a meta-model, a high-level conceptualization or an independent model of various agent systems that developers can use to model, extend and instantiate their specific agents.
This approach has the defect of being limited only to the three dimensions of MAS which have been cited above.

2.3.3 Semantic approach of negotiation

The approach developed in [V.Tamma *et al.* 2005] applies the semantic web technology to automate the negotiation between the agents. It is based on the definition of a negotiation ontology shared by the agents. The definition of this ontology is centered on the work in the electronic commerce field. From a representational point of view, the proposed ontology describes the negotiation protocols according to their process, participants and roles, negotiation rules and the purpose of the negotiation. As the authors point out, a protocol is therefore seen as a process. At the operational level, protocol behaviors are described using PSL8 (Process Specification

Language). This ontology has the defect of being limited only to negotiation protocols.
Indeed, it does not cover all the concepts necessary for the definition of a protocol (objective, parameters, exchanged messages). The representation is also abstract. From an execution point of view, the authors outline a possible execution model and some recommendations or requirements without a complete operational system. Indeed, it is difficult to ensure the automatic negotiation on the sole basis of an ontology without integrating a prior in the agents of the minimal knowledge on the protocols.
The point is to incorporate the knowledge about the protocols into the agents and to create subsequent matches between this internal knowledge and the negotiation ontology. This matching between the two representations requires translation mechanisms within the agents.
Finally, it should be noted that this work does not address the problem of the automatic selection of trading protocols.

2.3.4 Multiagent platforms

Many multiagent platforms support the interaction protocols to encode the protocols that enable the interaction between agents.
JADE [Home 2004] is a community platform distributed in open source. It supports standard FIPA interaction protocols. From the point of view of representation, JADE offers neither a model nor a language dedicated to the description of the protocols. Nevertheless, most FIPA protocols specified in Agent Unified Modeling Language (AUML) are available on the platform. This platform makes it possible to implement all the specificities of the AUML [Yves 1995] formalism: roles, performatives for message exchanges, cardinalities of the messages and conditions on the sequence of messages.
The platform offers no way to move from modeling to implementation. In order to implement a protocol, it is necessary to create a program for each role using the Java language by re-using the dedicated classes to the protocols. From an execution point of view, the roles are statistically and manually encoded in agents. To make the agents playing new roles, the developer must stop, reprogram, and restart them. The roles are performed in a distributed manner on the different participating agents and in a peer-to-peer mode. The reliability of the execution is not guaranteed in the absence of transformation between the specifications and the executable representations. For the protocol selection, it is the MAS designer who has to assign the roles to the agents manually.
Finally, it should be noted that although JADE supports the protocol execution, it has no specific tools or dedicated systems that cover the entire protocol life cycle.

2.4 Discussion

After having studied a panorama of works relating to engineering MAS, we observed that some problems were treated and that some specific software solutions already exist. However, there are still some shortcomings that we feel are important, such as the lack of a comprehensive approach to the entire adaptive MAS development cycle, from the specification of the ambient

environment to the use of a MAS.
We can note the following shortcomings:

- Coverage of requirements at the representation model level is not systematic. However, we must recognize that the model proposed by [W.Bouaziz & E.Andonoff 2015], offer an adequate power of expression since it makes it possible to take into account the majority of the concepts defining a protocol and the proposed system brings solutions to the problem of interoperability.

- The dynamic execution of the protocols is partially covered in the work of [V.Tamma *et al.* 2005] and [A.Freitas & R.Bordini 2017]. Given the complexity of this dynamic execution problem, the proposed solutions by some works are usually ad-hoc. Automatic protocol selection is not supported in all research.

- The system proposed by [A.Freitas & R.Bordini 2017] suffers in particular from the absence of tools relating to the different phases of the life cycle protocol. When an ontology is used for a MAS modeling, only a part is modeled, such as the environment or the organization.

The existing approaches depicted in table below 2.3 suffer from the absence of a global approach grouping the complete development cycle of a MAS; from the specifications of the environment, the agents and their interactions, to the deployment and the use of the MAS. The major lack resides in the fact that very few works consider modelling agent interaction protocols in order to automatically embed an adequate interaction protocol in each agent of the MAS. Indeed, we think that the adaptation ability of a MAS can be greatly improved by being able to adapt the interaction protocol in a dynamic way from the design step to the runtime step. In order to address this shortcoming, our approach, which is described in the following part, is based on ontologies and software engineering methods.

System	Representation	Selection	Dynamic execution
Protocol management system	Treaty	Treaty	Semi-treaty
Semantic approach of negotiation	Treaty	Untreated	Semi-treaty
Multiagent platforms	Untreated	Untreated	Untreated
Modeling, programming and verifying MAS	Treaty	Treaty	Treaty

Table 2.3 – Comparing related work of engineering MAS

2.5 Self-adaptive multiagent systems

Self-adaptive systems are capable of modifying their runtime behavior in order to achieve system objectives. Unpredictable circumstances such as changes in the system's environment, system faults, new requirements, and changes in the priority of requirements are some of the reasons

for triggering adaptation actions in a self-adaptive system. To deal with these uncertainties, a self-adaptive system continuously monitors itself, gathers data, and analyzes them to decide if adaption is required.
Among several existing definitions for self-adaptive system. Here, we enumerate only some of the most commonly accepted definitions.

Definition 1: Self-adaptiveness is the ability of the whole MAS to dynamically adapt its behavior to the execution context. This is done by each individual agent, through the dynamical execution of its own capabilities, according to a shared plan and to contingent perceptions [L.Sabatucci *et al.* 2018].

Definition 2: A self-adaptive system is a system with the ability to autonomously modify its behavior at run-time in response to changes in the environment [C.Mario & V.Betti 2014].

Definition 3: A self-adaptive system evaluates its own behavior and changes its own performance when the evaluation indicates that it is not accomplishing what the software is intended to do, or when better functionality or performance is possible [Salehie & Tahvildari 2011].

The challenging aspect of designing and implementing a self adaptive system is that not only must the system apply changes at runtime, but also fulfill the system requirements up to a satisfying level. Engineering such systems is often difficult as the available knowledge at design time is not adequate to anticipate all the runtime conditions. Therefore, designers often prefer to deal with this uncertainty at runtime, when more knowledge is available [Mahdavi-Hezavehi *et al.* 2017].
In our research, we view MAS engineering as a further abstraction of the object oriented paradigm where agents are a specialization of connected objects. Instead of simple objects, with methods that can be invoked by other objects, agents cooperate with each other via interaction protocol and act proactively to accomplish individual and system wide goals.
On the other hand, autonomic computing is a field of computer science aiming at building systems that are able to automatically and autonomously adapt their own structure and behavior in response to changes occurring in their operating environment. This concept is known as self adaptation [J.Oukharijane *et al.* 2019].
Self-adaptive software is capable of evaluating and changing its own behavior, whenever the evaluation shows that the software is not accomplishing what it was intended to do, or when better functionality or performance may be possible.

2.5.1 ADELFE method

ADELFE [C.Bernon *et al.* 2003] is a method dedicated to the development of adaptive MAS. The name “ADELFE” is the French acronym for “toolkit to develop software with emergent functionality” (Atelier pour le DEveloppement de Logiciels à Fonctionnalité Emergente). While ADELFE is not the only method devoted to guide the design of a MAS, it is the only one specifically tailored for adaptive MAS.

The ambition of ADELFE is to provide be-all and end-all method to guide engineers during all the phases of the design of an adaptive MAS, from the high-level requirements to the "nuts and bolts" implantation details. This ambition was the driving factor for multiple projects with the objective to improve or complement ADELFE with additional tools, such as the Make Agents Yourself (MAY) framework [Noel 2012], used to automatically generate agent architecture implementations.
However, as for most general engineering methods, a current limitation of ADELFE is that it only provides high-level guidelines concerning the behavior and architecture of the agents, staying at a general, abstract level. This current limitation makes difficult for a non-expert in adaptive MAS to actually provide an adequate instantiation for the problem he wants to solve.
It is the same analysis in [E.Kaddoum 2011] which led the author to prone a specialized variant of the method containing additional guidelines and tools for applying adaptive MAS in the context of problem solving.

2.5.2 Life Cycles for MAPE-K

To achieve the self-adaptation, several solutions have recommended the IBM's MAPE-K approach [IBM 2006], which is the de facto reference model to design self-adaptive software in the context of autonomic computing.
This approach advocates four functions including (i) Monitor (M) for collecting data about the managed system and its environment from sensors, filtering and aggregating them into symptoms, (ii) Analyze (A) for analyzing the symptoms to detect if changes are required, (iii) Plan (P) for constructing the actions needed to resolve detected changes and (iv) Execute (E) for applying the actions required to adapt the behavior of the managed system using effectors. These four functions (MAPE) handle and share Knowledge (K).
Another interesting contribution [Z.Pang & Z.O'Neill 2019] introduced an adaptation engine for self-adaptation of process instances at run-time. The adaptation engine is based on the MAPE-K approach. It introduces three types of agents, the agents monitor, adapter and executor that implement the different steps of the approach.
The agent adaptor implements decision making for adaptation, which is driven by goal and business rule analysis. It also implements the operations possibly required for process adaptation, in two steps:
first, it selects existing activities in the process repository or it uses planning technique for reconfiguration of activity coordination analyzing pre-conditions, post-conditions, interdependences between activities;
second, it uses business rules to define the required operations for process adaptation.

2.5.3 self-adaptive business process

Several contributions [R.Seiger *et al.* 2016] [R.Seiger *et al.* 2017a] have
suggest the self-adaptation of process instances based on their goals. More precisely, the authors proposed a framework for enabling self-adaptive business process in cyber-physical systems (CPS) based on the MAPE-K approach. The Monitor function collects context elements from

the physical world related to task goals. These context elements are then analyzed to check for CPS consistency after task execution with respect to task goals. In case an inconsistency is detected, i.e., a task goal is not satisfied, the task instance is (i) adapted by replacing the resource involved in task execution by another and (ii) executed afterwards in order to try to restore CPS consistency and continue with process execution as planned.

2.6 Conclusion

Diagnosing interaction protocol in MAS is a crucial task to engineering self-adaptive and cooperative multiagent systems for ambient assistive applications to assure the adaptation operation of a system.

In the first part, we have presented a state of the art on the Multiagent systems approaches for ambient intelligence. In the second part, we have presented a state of the art on the representation and implementation of interaction protocols. And we have come to the realization that a reusable representation of interaction protocols is crucial to facilitate the developer's task. One of the most important and principal qualities of MAS is the communication between agents representing a capital mechanism inside agents community. The existing approaches have a significant limitation for adaptability. The changing nature of networks of connected objects in a personal assistance system is problematic. The system must be able to adapt the agents created, more precisely, their interaction protocol according to the situations and the context of the application. In addition, a library (onotology) of interaction protocols is provided that can be used simply by the instantiation of the needed parameters.

In the last part, we have studied existing methodologies. We have observed that some problems have been addressed and that some specific software solutions already exist. However, we were able to notice several shortcomings which seem important to us: the absence of a global approach for the development of protocols, the failure to take into account the automatic selection of protocols according to the situation, the problem of dynamic engineering of protocols, and the lack of concrete tools dedicated to the development of the MAS.

Further chapters explain the main point of our thesis which consists of a methodology for protocol MAS modelling and automatically generating adaptive and self-adaptive MAS. Our research combines MAS with an ontology perspective for building self-adaptive MAS.

Part 2

Contributions

PART 2 presents the contributions of this thesis. This part focuses on presenting our newly developed approaches solving the adaptability of cooperative multi agent systems for ambient assistive applications.

The first chapter present a usage scenario for robot localization. Secondly, the next chapter investigates new approach for generating automatically an adaptive MAS for ambient assistive applications (GAAMAAA) in open environment.

GAAMAAA focuses on the design and implementation of the interaction protocol ontology and the creation steps of the MAS, first steps towards automating the interaction of the agents.

Finally, the last chapter deals the effective performance of GAAMAAA which instantiate localization robot scenario based on sensitivity analyses method.

Chapter 3

Robot localization scenario

3.1 Introduction

We relied on a usage scenario extracted from the Coalaa [N.Abchiche-mimouni *et al.* 2016] project in order to present the questions that our thesis attempted to propose answers for.
This scenario will play, throughout this thesis, the role of common thread to guide our reflection.

3.2 Scenarios description

The considered scenario consist in a variety of situations where an alarm has occurred (The scenario has been determined in cooperation with the remote monitoring center SAMU-92, which depends on Public Paris Hospital).
An alarm can be triggered by a device worn by the person or the sensor network of the ambient environment. The robot, thanks to its ability to move, helps to confirm and evaluate the severity of the alarm by cooperating with the COs.
The robot begins by searching for the person and then provides an audiovisual contact with a distant caregiver. That way, the distant caregiver is able to remove the doubt of a false alarm, to make clear the diagnosis and to choose the best answer to the alarming situation. It is important to note that the embedded device monitors the physiological parameters and the activity of the person. The originality of the proposed approach is that the robot tries to take advantage of ubiquity. The robot autonomy is obtained by a close interaction with the ambient environment (AE). So, the services the robot can bring to the user are directly related to the effectiveness of the robot mobility in the environment. Before providing a service to the person, the robot has to locate itself by interacting with the AE.
In these scenarios an ethical dimension, represented by the Level of Intrusion (LI) of the system, has been introduced to preserve the privacy of the person. The LI is defined according to the degree of freedom of the system regarding to its actions. For instance: maximal distance allowed between the robot and the person, activating a camera, switch on a light and so on. The LI is supposed to be minimal except in a case of an emergency.
The evolution of the LI based on the degree of urgency of the situation or not and the availability of the different communication devices of the ambient environment are particularly targeted.

3.3 Robot localization task

Using a robotic assistant for the task rather than a simple set of fix cameras in all rooms is an advantage in two cases: i) the assistance is only needed for a limited period such as convalescence period or ii) the residence is composed of too many rooms for example nursing home. Another advantage is that the quality of the image and the sound are better. The goal of the robot is to autonomously move to the person in case of an alarm and then to provide an audiovisual contact with a distant surveillance center.
Figure 3.1inspired from this work [N.Abchiche-mimouni *et al.* 2016] shows a robot in the person's home; the patient has fallen. To move towards her/him and to guide its camera to the remote caregiver, the robot has to be located first. A visual contact will help the remote care-

giver to perform a correct diagnosis of the situation.
If the robot is located at P1 position, then its mobile camera can identify the visual marker

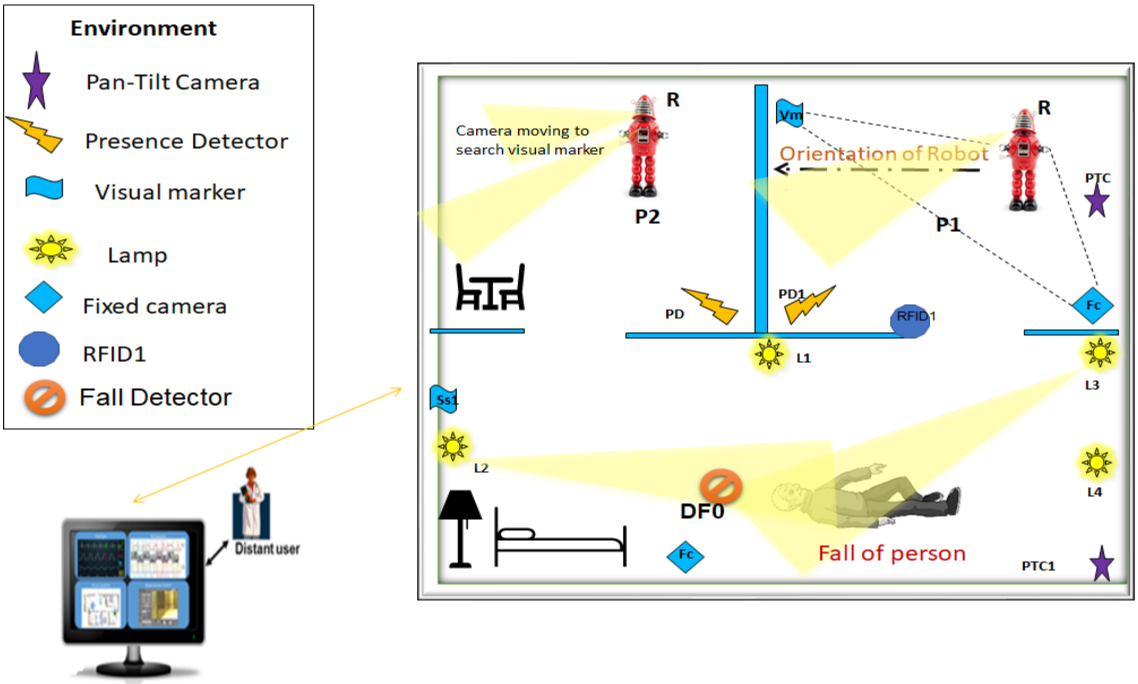

Figure 3.1 – A person falls scenario

Vm. With further information from a fixed camera environment, the robot manages to locate itself by a mean of an adequate localization algorithm. The direction taken by its mobile camera that detected a visual marker also allows the robot to know its orientation relative to a fixed reference in the environment. This information can also be inferred from previous values using odometry on the one hand and its linear and angular speeds on the other hand. It is thus easy and straightforward to identify and understand that the more information you have the better the accuracy of the location of the robot is.
If the robot is in P2 position, it has no marker on its visual field and has no element enabling it to locate itself. It then uses two different strategies to find a visual marker. Either its moves randomly or turns its pan-tilt camera. In two cases, it is necessary that the intrusion level of the system permits it. It can also query the detectors of presence to learn about the place in which it has been seen lately. In the case of several conflicting reports, it will be decided according to the data freshness criteria, or according to the consistency with the data criteria already available thanks to the sensors of the robot.
This simple scenario shows that robot localization is a complex task and there is no evidence for an approach that could be able to choose the relevant interactions between the robot and the AE. The difficulty lies in choosing the most relevant criterion to be considered first: is it the closest CO, the most accurate and or least intrusive? The problem analysis suggests that depending on the context, the criterion to consider is different. As the context itself is dynamic and difficult to predict, a centralized algorithmic solution is to be excluded. What is required is an approach

that can adapt the selection and the use of criteria based on the context and the choice of LI aligned with the level of urgency.

3.4 Scenario formalization

Typical scenarios that can benefit from our self-adaptation approach are those characterized by the following three characteristics:

- the dynamic environment can present unforeseen events, which require dynamic modification,
- the dynamic changes need to be slightly flexible: in each new phase, other agents can be introduced or some of the old agents are no longer involved (temporarily or permanently),
- these dynamic changes need to be non intrusive, by considering a dynamic LI which depends on the degree of urgency of the situation. The higher is the degree of urgency, the higher is the level of intrusion.

In order to give a preliminary idea of our system, we simplify the above scenario in the case of an environment detailed in Figure 3.1 which has three CO: R(Robot), FC (Fixed Camera) and SS (Sound sensor). Then we show how self-adaptation works when a CO incapacitates. The robot requires the FC and the Ss to collect more information to locate itself when an alarm occurs, suspecting a person fall.

3.5 Scenario parameters

The scenario is modeled by a triple $\sigma < t, c, f >$ where:

- t $\in$ T; where T is a set of tasks labels (localization robot, enlighten, cognitive stimulation...);
- c $\in$ C; where C is a set of criteria (ability of communicating object, efficiency, time constraint, neighborhood...);
- f $\in$ F; where F is a set of influencing factors (level of intrusion, degree of urgency...).

3.5.1 Measure

In this part, we propose to draw up the different measures adopted for the evaluation in the localization scenario. The purpose of this section is to provide possible interpretations for each of these measures.
The following measures come from the field of personal assistance applications.

- Degree of urgency measure: the degree of urgency (DU) is a necessary constraint for the intervention of CO. It is an independent parameter sensitivity [Z.Pang & Z.O'Neill 2019]; we used this variable to check the response time of running task.

Parameters	Local	Global	Interpretation
DU		X	Time required for intervention: Task processing time
LI	X		Type of agent involved in : - a Coalition - a coordination process
LP	X	X	Related to the competence of agents. Related to global results provided by the MAS

Table 3.1 – Summary and possible interpretations of the interaction parameters

- Level of precision measure: the level of precision (LP) is an attribute resulting from the technical characteristics of the COs. We will opt for empirical and relative values by ordering the CO according to their level of precision on a scale going from the less precise to the most precise. Then, the value space is enlarged to allow the system to provide results of a precision better than that which would be provided by each CO individually. It is depending parameters sensitivity [Z.Pang & Z.O'Neill 2019]; we used this variable to select the competence of agent.

- Level of intrusion measure: the level of intrusion (LI) is an ethical constraint, the intrusion of an ambient intelligence capable of acting and perceiving in the presence of the person causes discomfort for him and his entourage. It is essential to limit this discomfort as much as possible. It is depending parameters sensitivity [Z.Pang & Z.O'Neill 2019]; we used this variable to select the agent entitled to participant role.

3.5.2 Interpretation of the measurements

Thus, we summarize through table 5.1 the different parameters associated to the interaction in the MAS. For each parameter we determine if it is a local parameter, that is allowing to characterize a CO, or a global parameter which characterize the entire MAS and the application [F.B.Hmida *et al.* 2015]. We also present the possible generic interpretations of these different parameters in the form of indications which will be developed later on based on a scenario for carrying out a specific task.

3.6 Scenario challenges

The first step on any robotic localization is to perceive the environment through communicating objects (sensors). In our scenario, it consists of acquiring context data from various sensors, from the robot or the smart environment. The objective is to gather all the data and maintain a context knowledge.

3.6.1 Unpredictable situation

Areas at risk of falling frequently occur in “risky” places such as kitchen,bathroom, in corridors, stairs or even in the garden. Therefore one cannot determine position of the person when falling. From the acquired context data, the robot knows that the person has fallen and take the move to assist the person and/or alert the medical staff. For this reason we need more and more the localization robot scenario cooperated with connected objects.
We propose an enhancement to the vision of position of person according to the context knowledge.

3.6.2 Dynamic environment

A dynamic environment is characterized by the uncertainty of the environment that limits the ability of system to make decisions.
In ambient environment, uncertainty is a serious issue. In fact, we should expect that sensors will provide non-perfect data. For instance, a thermometer may not be accurate or a motion sensor may be intellectively triggered. In such cases, uncertainty can lead to the production of uncertain context data. Subsequently, this may cause problems of bad surveillance.
The first step on any robotic localization is to perceive the environment through communicating objects (sensors). In our scenario, it consists of acquiring context data from various sensors, from the robot or the smart environment. The objective is to gather all the data and maintain a context knowledge.
In a context of a robot location in ambient environment we identified four essential dimensions:

- Freshness: data is outdated. For example, if a sensor sends an event asserting the user is in room B while he has just left for room B.
- Precision: data is correct yet inexact. For instance, a motion sensor as RFID only detects a movement, it is imprecise compared to a camera that detects movement and identifies a user.
- Contradiction: two pieces of data provide contradictory information. For example, when the user is between two rooms and detected by sensors of both rooms.
- Availability : the communicating objects may also be unable to provide the required service any longer, they may provide it with poor quality of service or they may be temporarily unavailable.

3.6.3 Changing context

A lot of changes can occur within the environment in which the robot is deployed in. For example, if the robot is asked to fetch some medications and the user changes the position of the robot while it is reaching their location, it won’t be able to find them, thus failing to achieve its task. Another complication is the transition between day and night, for example at night more communication is needed, the light context is different and therefore additional requirements are required such as infrared cameras. The presence of different caregivers between the weekends and weekdays demonstrates another change of context.

3.6.4 Ethical and urgency factor

Ethical factor: Access to real time images: put cameras in every room is a very intrusive solution.A robot equipped with audio visual device, activated essentially if necessary seems the point of view ethical lighter. The robot can move closer to the person, the operator has a better view of the scene.
Urgency factor: Both context variability and missing data at planning may lead to failure situations that would drastically slow down the robot. This is not acceptable as the user health can be in the balance. For instance in the case of a fall of the user, the robot shall not waste time trying possible options to reach him/her.

3.7 Conclusion

As expressed in the previous section, each step has specific challenges for our context of a robot location in a smart home. Consequently, specific solutions must be proposed to each of these challenges in order to provide an efficient framework.
Adaptive systems [A.Eduardo *et al.* 2003] are known to meet this requirement. More precisely, adaptation features are inherent to MAS. So, our approach exploits the MAS adaptiveness potential to design a distributed system to deal, in a dynamic way, with scenarios such as the one described above. The adaptiveness is also needed to deal with dynamic addition and suppression of sensors. There is no deterministic algorithm for solving this type of complex and distributed environment. To solve these challenges, we propose a self-adapting architecture detailed in next chapter.

CHAPTER 4

GAAMAAA approach

4.1 Introduction

The objective of this thesis is to propose a computer architecture allowing a self-adaptive cooperation between the robot and the communicating objects which present in the environment such as sensors and actuators. One of the difficulties is that the robot and the surrounding environment are dynamic systems in the sense that, for example, ambient sensors can be available / unavailable, accessible / inaccessible. Further considering the fact that ambient sensors are generally unevenly distributed throughout the environment, it becomes difficult to design a deterministic approach capable of meeting the objectives of the scenarios for which the system is designed. It is why, we have turned to an approach with adaptation mechanisms allowing not only to take into account the dynamic aspect of the environment, but also factors like the degree of intrusion of the system on the privacy of the person and the precision of the measurements of the sensors.

In the context of the scenario described in the previous chapter concerning the removal of doubt after an alarm, the adaptation mechanism must take into account two constraints. The the first is linked to the obligation of result. The adaptation mechanism must respond to various or even unforeseen situations so that the robot ultimately succeeds in its mission, moving from point A to point B, if possible taking into account the degree of urgency. The other constraint is of an ethical nature, the intrusion of an ambient intelligence capable of acting and perceiving in the presence of the person causes discomfort for him and his entourage.

Thus, this chapter begins with a description of our approach based on MAS and the principle of the interaction protocol between agents. This will help highlight the adaptation mechanisms underlying such approaches. We will then describe the general scheme of our GAAMAAA architecture (Generating Automatically an Adaptive Multiagent systems for Ambient Assistive Applications).

Then, we present a dedicated ontology ambient assistance designed to model the body of knowledge useful for the representation of the environment, and for the operation of the MAS. Thus, we present an ontology dedicated to the interaction protocol.

The latter is described in detail by illustrating the interaction mechanisms for obtaining a task, on the scenario considered. The protocol for evaluating the approach with results is described in the next chapter.

4.2 GAAMAAA architecture

Our GAAMAAA approach proposes an architecture as an attempt to automatically generate a set of cooperative agents in order to provide a service adapted to a dependent, elderly, or a suffering person maintained at home. The latter is equipped with a set of COs and possibly with a robot. This work is in the continuity of the COALAA project [Nadia.Abchiche.Mimouni *et al.* 2013] in which a personal assistance ontology (AA) contains the information linking the services rendered to the person, a description of the home and the installed COs as well as the knowledge about the person's profile. A cooperative and adaptive MAS has been developed to provide a person-friendly service. In this system, each agent encapsulates a CO. The objective is to contribute to the improvement of the adaptation capacities of COALAA by offering a dynamic and automatic generation of the agents and their interaction. The architecture of our contributions is represented in figure 4.1. It is organized in three levels: (1) Ontology level, (2) Configuration level and (3) Application level. Each level composition will be detailed below.

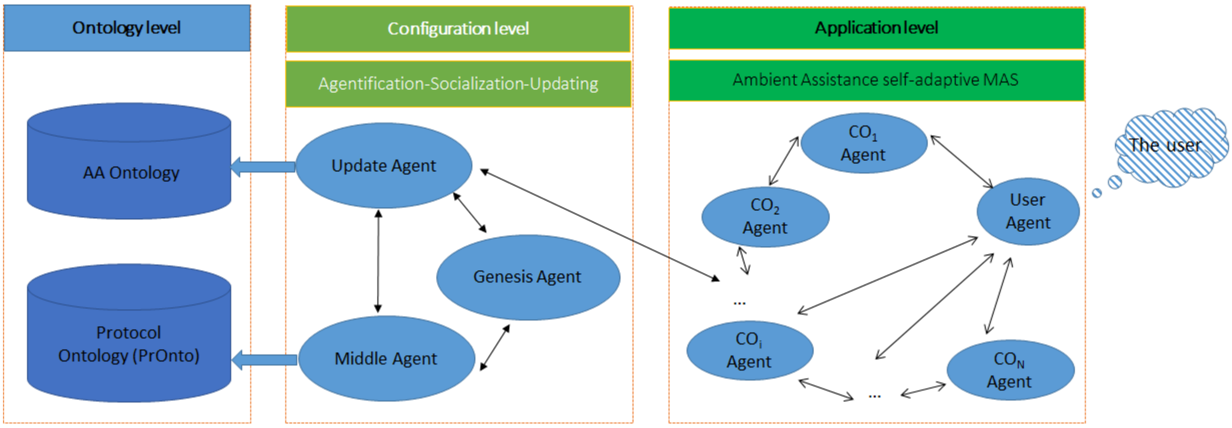

Figure 4.1 – GAAMAAA Architecture

4.2.1 Ontology level

We have used two ontologies :

- An ambient assistance domain ontology (AA); this ontology contains all the information relating to the services that we want to give to the older adult as well as those necessary for the different actions that the robot could perform. It also includes both the description of the domain components that are people, home and connected objects (robot, sensors and actuators) but also the tasks and services.

- An ontology of interaction protocols; this ontology is considered as a model defining the structure shared by all the protocols. More precisely, this ontology makes it possible both to describe the profile of the protocols as well as their control structure. The profile defines what the protocol does, describing. The purpose of the protocol, specifying the protocol category, the problem types resolved by each protocol, and specifying some of its static properties. This is the profile of the protocols that will be used to perform their dynamic selection. The control structure of a protocol defines its behavior and its functioning through the description of the actions that it supports and the scheduling of the messages that it allows exchanging.

4.2.1.1 Protocol Ontology (PrOnto)

The concept of interaction protocol is an effective way to structure and to organize the exchanges between the agents of a MAS. We have proposed an ontology for modeling interaction protocols

(PrOnto) in a generic way.
The ontology we propose is considered a reference model for the description of interaction protocols. It thus provides a declarative and explicit representation of these protocols. We have listed all of the terms or concepts relating to the protocols. For this, we have studied some protocols existing in the literature such as the Contract Net protocol, the Iterated Contract Net, the auction protocols, the coalition protocol, and the negotiation protocols. The goal of the work [W.Bouaziz *et al.* 2009], [V.Tamma *et al.* 2005],[T.Jarraya & Z.Guessoum 2007] studied in section 2.3 is to facilitate the use of interaction protocols, without requiring a great effort from the developer.The proposed ontologies can be classified by type of protocol (coordination, negotiation). No solution has proposed an implementation of interactive agents with dynamic reuse of interaction protocols. The PrOnto ontology groups together properties and behaviors common to several types of interaction protocols described by categories concept. Our goal is to provide an ontology for the reuse and dynamic execution of interaction protocols through the Role concept which instantiates each role of an agent in a protocol in a personal assistance environment. Figures A.5, A.6, A.7 and A.8 gives a graphical representation of the protocol ontology using the concepts of the OWL language, namely the notion of class, objectproperty, datatypeproperty etc.
Properties of PrOnto

To fix the terminology, from the existing works [W.Bouaziz *et al.* 2009], [V.Tamma *et al.* 2005],[T.Jarraya & Z.Guessoum 2007], we have considered that an interaction protocol is characterized by the following fundamental properties:

- Category: function or purpose of the protocol; negotiate, cooperate, contract or auction. A protocol consists of a set of elements that define it: rules, roles, constraints and messages. Let's detail each of these elements.
- Role: abstract behavior expressing the function of a participant involved in a protocol.
- Action: Internal operations performed by a role, whether communication action (sending/ receiving a message) or decision-making (ex. choice of an interaction duration).
- Domain ontology: vocabulary used in the content of the messages.
- Message: informational content exchanged by the participants.
- Constraint: conditions to be satisfied by an agent to play a role in a given protocol:
  - Distribution Constraint: the agent conditions of use and role are based on internal criteria related to the agents (capacity) and on external criteria related to the context (levels of intrusion, precision).
  - Behavioral Constraint: conditions on the performance of a task by an agent (timeout, scheduling).

- Rules: rules which control the structure of the logical sequence of actions and exchanged messages.

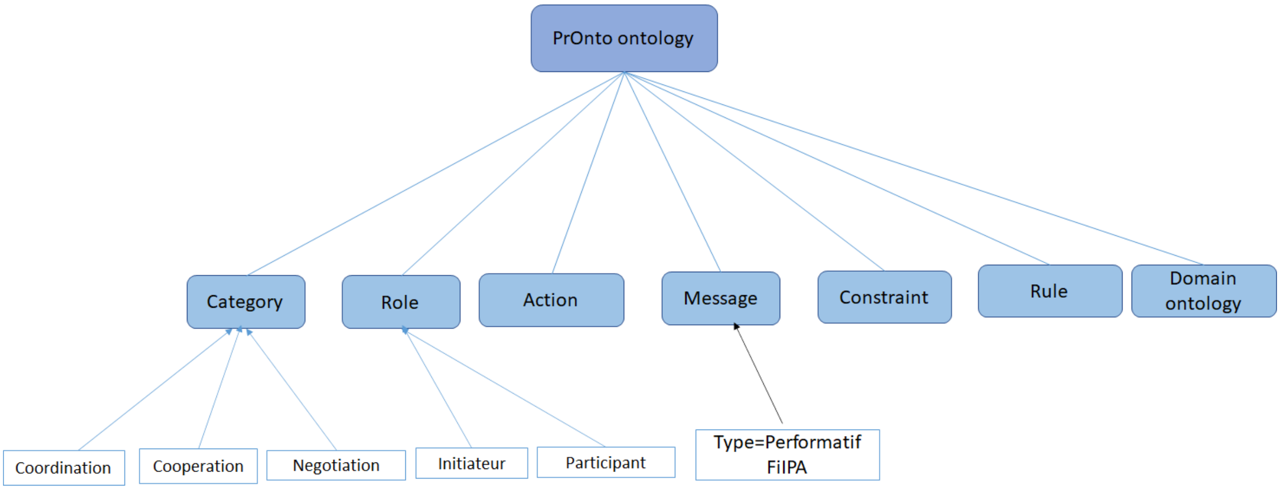

Figure 4.2 – Protocol ontology (PrOnto)

4.2.1.2 Domain knowledge modeling : Ambient Assistance Ontology (AA)

We opted AA (ambient assistance) ontology which has been developped by [Nadia.Abchiche.Mimouni *et al.* 2013]. This knowledge base includes the persistent information useful for generating the MAS. This information is classified representing the main objects of the application; the home, the connected objects, the tasks and the people likely to intervene in the application, outside the assisted person (caregivers, medical staff ...). Some of this knowledge is used to initialize the MAS.
<u>Properties of AA</u>
The AA ontology contains four categories of information related to the application domain 4.3:

- The Home category for defining the structure of the environment.
- The Communicating Object (robot, sensors and actuators) category for describing the characteristics (accuracy, orientation, position...) of the COs and their operating mode (effect).
- The User category for defining the user's profile and preferences.
- The Task category that puts together the tasks and services the system is able to provide. Some of this knowledge is used to initialize the MAS.

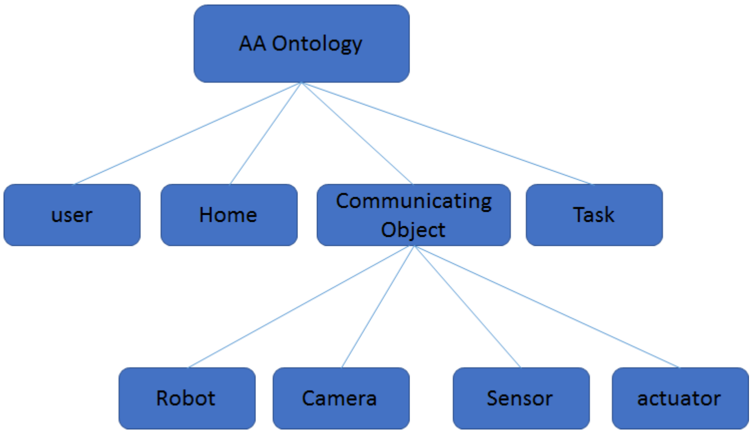

Figure 4.3 – Ambient Assistance Ontology (AA)

4.2.2 Application level

The developed MAS relies on the information that it receives from a knowledge base and the sensors of the environment (COs), to make a reply when a request for a service (lighting at a specific location of the residence or the location of a robot) is issued from the user.
The originality of the generated MAS is to be endowed with the capacity of selfadaptation. It gives it dynamism and flexibility for understanding a variety of situations and problems. More specifically, it is to adapt to different usage scenarios which take into account the context of the person and his/her profile (preferences). Each agent is associated to a specific CO and defines locally and proactively the time and the manner of the contribution to the service that is required to the person. The methodology for an automatic generation of such a MAS is described hereafter.

4.2.3 Configuration level

Automatic generation of a self-adaptive MAS
The process of generating an adaptive MAS follows 3 main steps: (1) Agentification step, (2) Agents creation step and (3) Socialization step. The Agentification is the process that allows you to define the elements of the field of application which will be ratified in the form of agents. This step is to Create agents in a platform multi-agents . The step of Socialization is to embed at the level of each agent a protocol that allows him to interact with other agents.
The first two steps are performed by an agent called Genesis Agent, responsible for the creation of agents from information from the AA ontology. During the modeling of the AA ontology, it integrates information allowing to select the COwhich will be reified in the MAS. This information is used by the two algorithms below to generate the OC-agents (agents that model each an OC).

4.2.3.1 Agentification and agents creation steps

The Agentification step is the process that allows to define the elements of the application domain which will be reified in the form of agents. The agents creation step consists in creating the agents entities in a multiagent platform. These two steps are performed by a configuration agent called GenesisAgent, responsible for the creation of the agents. The AA ontology contains prior information allowing to identify the CO (concepts) which will be reified in the MAS as agents. This information is used by the two algorithms below to generate the CO-agents (agents that model the COs). The Algorithm 1 takes as input the AA ontology in the form of a OWL (Ontology Web Language) file and retrieves the list of the concepts whose property "Agentifiable" is assigned to true. It provides the output in a form of a set of Java classes corresponding to the extracted concepts. The attributes of these concepts are the attributes of the Java classes and instances of the concepts are Java objects. The creation of the agents takes as inputs the outputs of Algorithm 1 and creates an agent for each instance. A particular agent named UserAgent is in charge of interacting with the user in order to submit the request to the system (as explained further).

Algorithm 1 Agentification

```
Input: AA Ontology
Output: MAS
  //start with declaration of variables
  Var : integerCounter = 0, listClassList[]
  Begin
  // Read ontology owl (AA)
    Ontology_read( OWL File )
  while (Not end OWL File) do
    ClassList.add(OntologyAA.class);
  end while
  while (Counter < ClassList.size()) do
    Counter++
    if (Counter.ObjectProperty <> "Agentifiable") then
      ClassList.Delete(Counter)
    end if
  end while
  // Generate Java classes for each connected objects with these attributes
  Java-Classe =GenAutoClasses(ClassList, Dataproperties, ObjectProperty)
  if (Dataproperties == "Agentifiable") then
    Create − Agent − CO(Java − Classe)
  end if
Return ExecuterJADE();
End
```

4.2.3.2 Socialization step

The socialization step is to embed in each agent a protocol that allows the agent to interact with the others. The stage of socialization is carried out by another configuration agent named MiddleAgent, which can use the algorithm 2 or the algorithm 3 according to the chosen interaction protocol. The Algorithm 1 is related to an Interactive Contract Net protocol (ICNP), while Algorithm 3 is dedicated to the coalition-based protocol. The most important part of these two algorithms is the role extraction and the instantiation of the protocol.

ICNP Algorithm

Algorithm 2 (named AutoProtocol-ICNP) has three inputs: the URL of the Protocol ontology, the agents list and the user request. It starts by role extraction, by selecting information from the protocol specification. For example, to instantiate initiator role, the "Instance_of_Role_Initiator(PrOnto)" function is used.
A generated Java class represents the instance of the role, where roles are embedded as agent behaviors.

Algorithm 2 Auto-protocol-ICNP

```
Input: List agentList[], Url:ontology PrOnto, String Request
    Begin
    Integer i=1
    Integer Random = Random-Selection(agentList[])
    // Creates a new ICNP: Initiator-Role
    Agent − Initiator = Instance_Of_Role_ICNP_Initiator(PrOnto)
    //Parameters: agent, msg - initial message
    Add.Behavior(new Agent-Initiator(agentList[Random], ACLMessage CFP))
    while (i <> Random) And (Not End agentList[]) do
        Creates a new ICNP: Responder-Role
        Agent_Responder = Instance_Of_Role_ICNP_Responder(PrOnto)
        Add.Behavior(newAgent_Responder(agentList[i], MessageTemplatemt))
        i++
    end while
    End
```

Coalition protocol Algorithm

The Algorithm 3 (Auto-Protocol-Coalition) takes as inputs the URL of the Protocol ontology, the agents list and the user request. It provides, as a output, the instances of roles of coalition (behaviors for candidate and initiator roles) in order to deal with the request.

Algorithm 3 Auto-protocol-Coalition

```
Input: List agentList[], Url:ontology PrOnto, String request
    Begin
    Integer i=1
    while (Not End agentList[]) do
        // Creates a new coalition protocol: Initiator and Candidate role
        Agent − Candidate = Instance_Of_Role_Coalition_Candidat(PrOnto)
        Agent − Initiator = Instance_Of_Role_Coalition_Initiator(PrOnto)
        //integrate the coalition protocol role into the agents with the desired effect name and requested capacity
        Add.Behavior( new Agent-Candidate(agentList[i],String effect, Integer capacity))
        Add.Behavior(new Agent-Initiator(agentList[i],String effect, Integer capacity))
        i++
    end while
    End
```

At the end of the socialization step, the generated MAS is able to process a user request. The robustness inherent to MAS is achieved through the Update Agent whose role is to inform the Middle Agent (see Figure 4.1) when a CO is no longer in the system or if a new CO is added to the AA ontology. The next section illustrate the application level of GAAMAAA architecture.

4.3 GAAMAAA illustration

In order to illustrate our approach, we proposed a simple case study based on a scenario in which a remote user requests a service. This request, submitted via a graphical interface (Figure 4.4), is sent to the MAS in the form of a triple = [t, c, f] where t ∈ T (T is a set of tasks such as localizating, lighting), c ∈ C (C is a set of criteria such as precision, efficiency, time constraint, neighborhood) and f ∈ F (F is a set of influencing factors such as intrusion level, urgency degree). In the considered scenario, we have considered a precision equal to 50 degree (scaled from 0 to

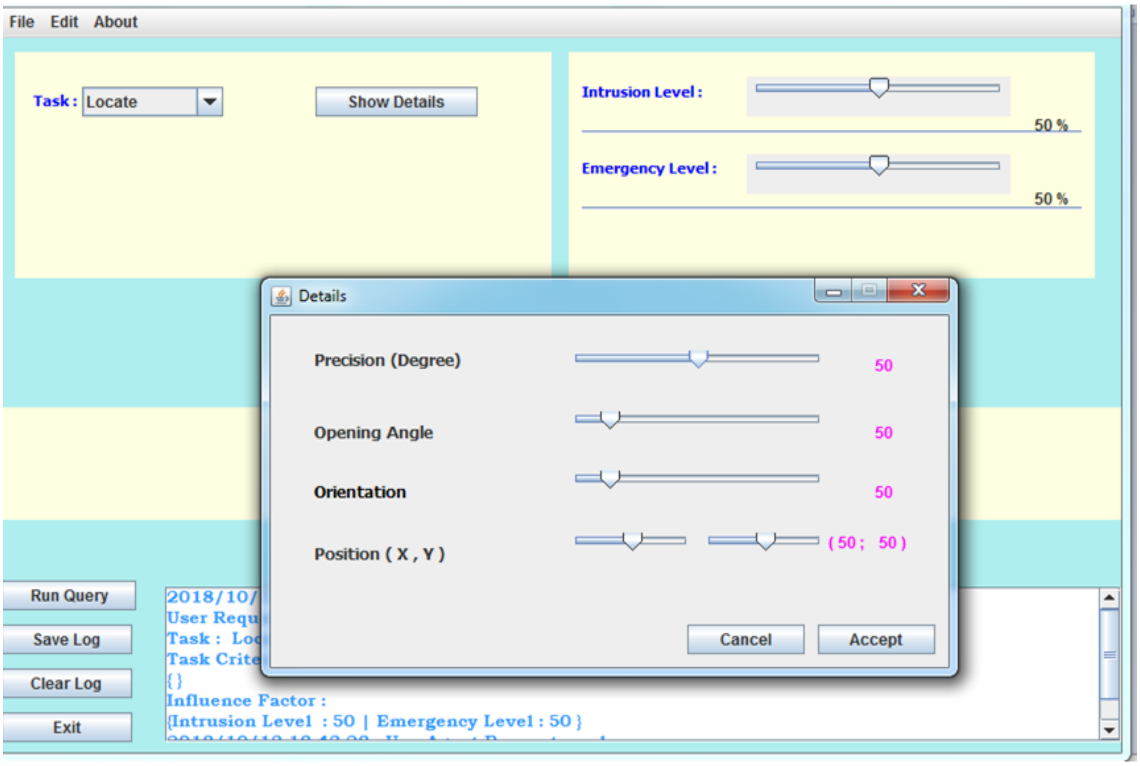

Figure 4.4 – Interface of service request

100), a level of urgency equals to 50 per cent and an intrusion level initialized to 50 per cent.

4.3.1 Agent and environment in MAS

We have opted for the design of an ontology adapts to the environment of the ambient assistance, inspires one developed by project Coalaa [N.Abchiche-mimouni *et al.* 2016].
It is to exploit the CO of the environment to provide a service (detection of falling, location...) adapted to the person. The environment is modeled by AA ontology. Each instance of CO have the property $Agentifiable$ represents a concrete agent in the MAS (see figure4.5).
Grant to each $Agentifiable$ instances an agent who carries the same attributes. Once the agents are created it loads the protocol (See figure below 4.6). Figure 4.7 illustrates in more detail how such example can be specified using the visualization provided by the Protégé ontology editor.

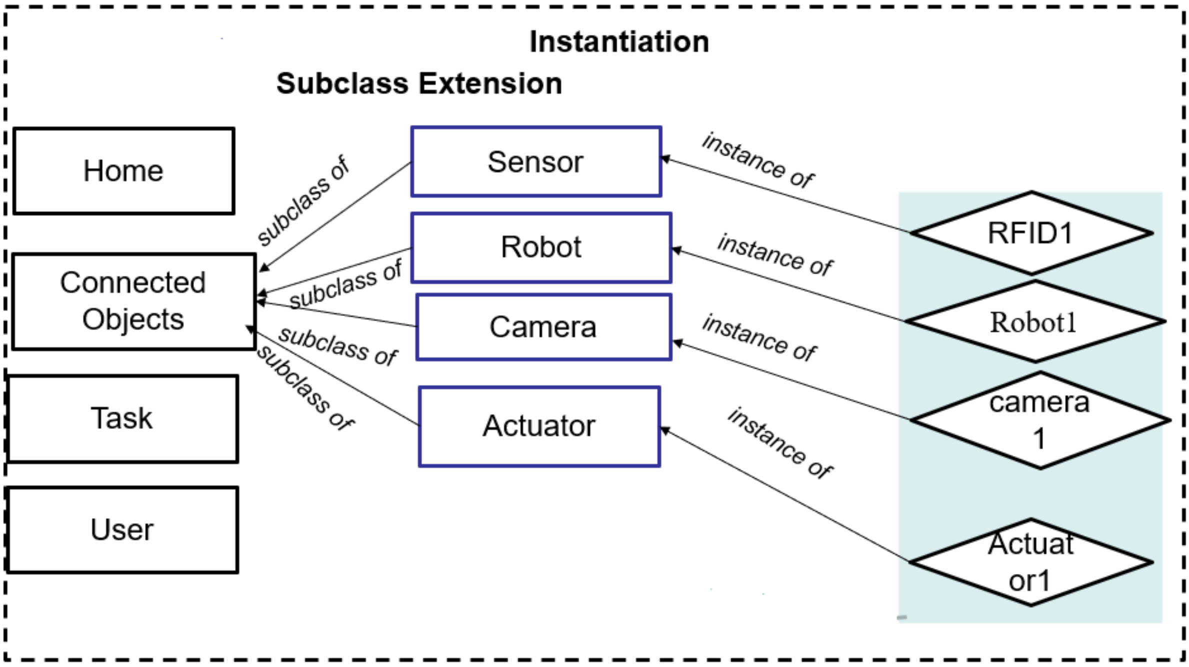

Figure 4.5 – Environmental of the ontology for MAS

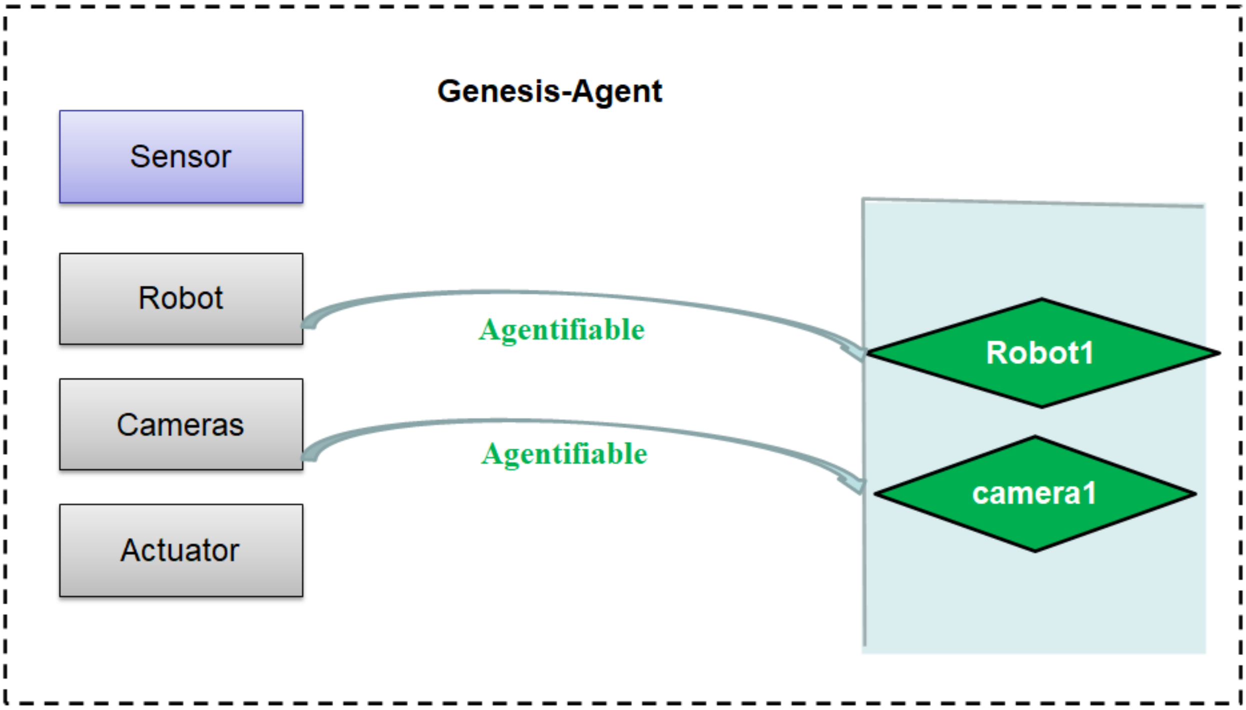

Figure 4.6 – Agent in MAS

Defining an individual instance of a class requires:
(1) choosing a class,
(2) creating an individual instance of that class and
(3) filling in the slot values.
For example, we can create an individual instance Robot1 to represent a specific type of Robot. Robot1 is an instance of the class Robot representing all Robot. This instance has the following slot values defined in Figure 4.7:

- Name: Romio
- Identify: 1
- MetaPropritiesAgent: True
- Objectpropritities: Agentifiable

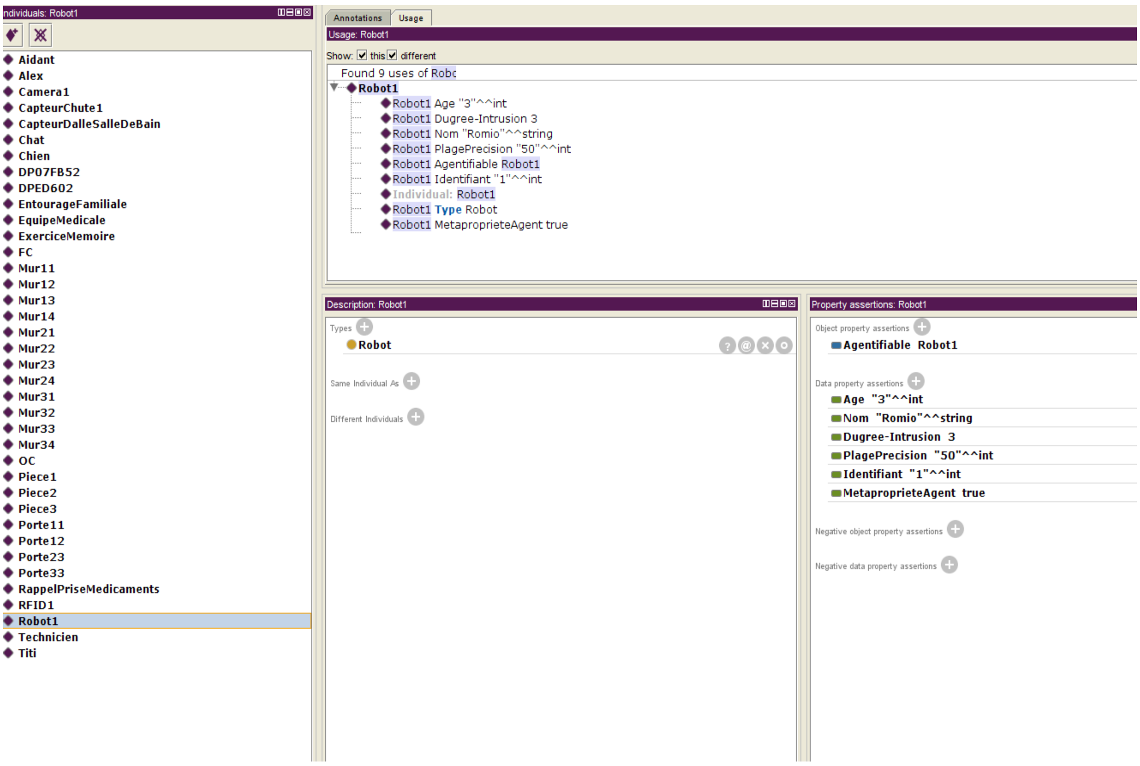

Figure 4.7 – Subclasses of communicating object with some instances

The right-hand side of Figure shows details about individuals and property assertions regarding the modeled MAS localization scenario. It illustrates the instantiation of concepts and properties that are asserted.

4.3.2 Interaction protocol in MAS

Figure 4.8 shows subclasses and instances to represent interaction characteristics of MAS. The diagram shows the instantiation of ICNP. In this protocol, agents can take two roles: manager or contractor. The manager is responsible for the supervision of the execution of a task and the processing of the results of this execution. A contractor proposes the manager to perform the task. In case of acceptance, it is responsible for the effective execution of this task.
The category of the protocol is given by the following three subclasses: Coordination, Cooperation, and Negotiation. Instances of these subclasses of coordination protocol (e.g., ICNP) define an assignment of that type mission to an agent. Two subclasses specify types of roles: Initiator and Responder. Instances of these subclasses, as Initiator1 and Responder1, define which agents are playing similar roles. The modeling also requires the creation of relationships among the individuals. Also requires the definition of other concepts such as AbstractAction that can be simple; sending or receiving a message or to Evaluate proposal sends by Responder as a CFP(Call For Proposal) message.
This example shows how to encode part of one possible strategy for modeling interaction characteristics of agent systems. However, other strategies are possible and would result in different designs and implementations for this scenario. Figure 4.9 shows a scenario where four CO

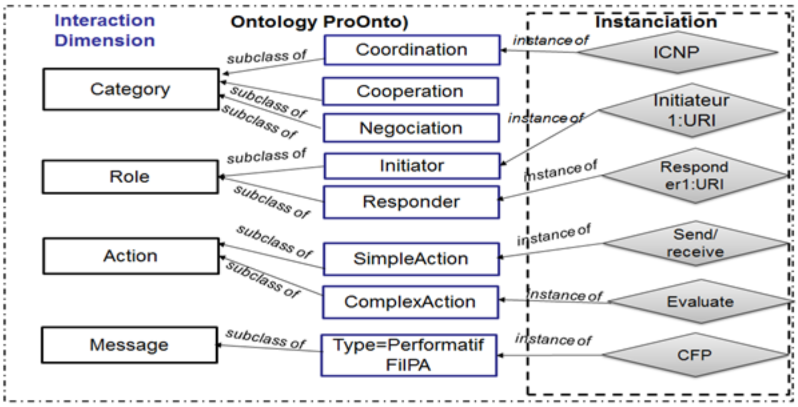

Figure 4.8 – Interaction protocol

agents have been automatically created on Jade platform [Home 2004], using the AA ontology and the embeddedness of an ICNP using PrOnto ontology to deal with a user request (localize the robot). In this scenario, four CO of the environment are used: a robot, a fixed camera, a presence detection sensor and a fall detection sensor. These four CO are encapsulated in four ambient agents (see red boxes). The initiator agent (Camera1) has received the desired effect from the interface agent and then broadcasts the request CFP (Call For Proposal) to all agents of the MAS. Each agent who receives the desired effect checks if its ability is adequate with the request. If yes, it sends an acceptance message labelled ”Propose” to be a responder. Such a message contains the capabilities of the agent. The initiator agent receives an acceptance answer, and chooses the abilities to consider for the localization task. Then, it confirms its acceptance by sending a message which labelled as ”AcceptProposal”.

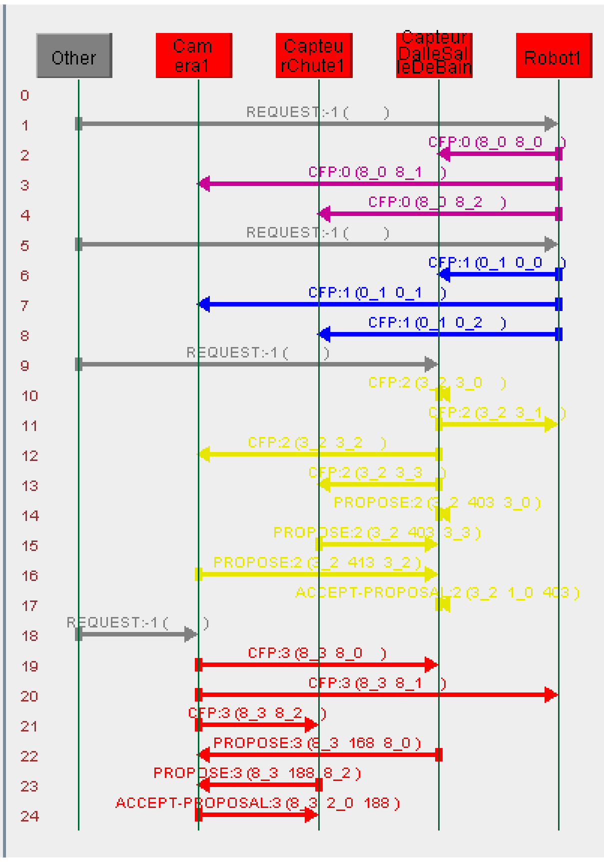

Figure 4.9 – Agent interaction diagram in Jade platform

4.3.3 Agents conformity

At this level, we have used our results in order to check the conformity of the built MAS to the ambient environment. Our test is based on the number of connected objects and the number of agents that are present in the MAS.
We considered a simple initial example with 4 instances of CO (sensor1, Robot1, RFID1 and camera1). Automatically we obtained 4 agents (each agent corresponds to a CO). The number of CO is upgraded to 7 and then to 14. In each case, the number of CO was equal to the number of created agents.

To ensure the synchronization between the ambient environment and the MAS, the MiddleAgent (see figure 4.1) manages the update of the AA ontology and the ambient environment and readjusts the MAS. This is either by providing the creation of new agents in the case of additions of new CO in the environment, or by removing each agent whose CO is out of service or has been removed from the environment.

4.3.4 Protocol performances

During the second set of experiments, the impact of the number of agents on the performances of the ICNP protocol is studied. The results represent average values for 20 runs with the same user request. The results are broken down into three categories (See Figure 4.10):

1. Memory usage: the memory rate used by each agent for the resolution phase is directly related to the number of formed coalitions.

2. Response time: the response time of the MAS corresponds to the average response times which are obtained after 20 executes of the scenario, while the number of agents varies in {4, 7, 14}.

3. Communication load: the total number of exchanged messages.

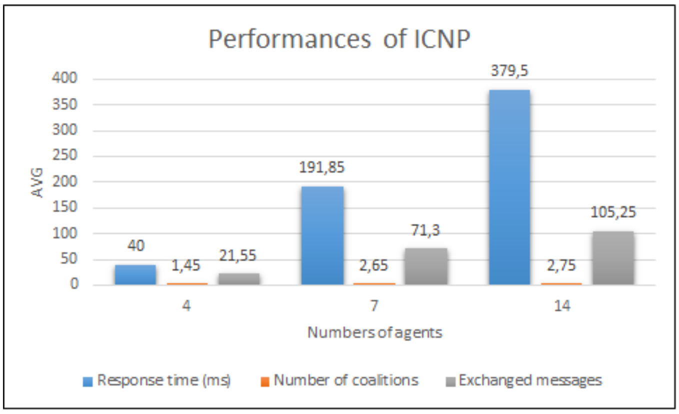

Figure 4.10 – *Performances of ICNP*

Note that the response time of the MAS impacts the number of exchanged messages. Hence, the sensitivity of the system to the number of agents is linear.

4.3.5 Quality assessment

The third kind of evaluation that is important to perform concerns the qualitative evaluation of the generated MAS. We focused on quality attributes that are relative to the interaction protocols. We proceeded to a state of the art, based on [L.Damasceno *et al.* 2018], in the domain of quality assessment of MAS and we have identified four minimal qualities that are relevant for a quality assessment of interaction protocols. These qualities are summarized in 4.1.

Quality	Description	Component	Metric
Performance quality	Suitable execution protocol.	Agents relationship	Effectiveness
	Suitable communication.	Agents Communication	Compliance
Reliability	Changes do not affect the MAS quality.	Changing of Parameters	Correctness
Agent quality	Responsibilities of roles represented.	Role Responsibilities	Completeness
	Sufficient skills for problem domain.	Agent Skills	Context coverage

Table 4.1 – Protocol quality

The column *Description* gives the meaning of the *Quality* attribute, while the *Component* column indicates the part of the MAS which is impacted by a quality. The *Metric* column gives a way of evaluating the degree of achievement of a quality and its adequacy to the environment where the MAS operates.
Effectiveness and Compliance metrics indicate the ability of the agents to achieve their goals through their relationships. These can be measured by evaluating the correlation between the protocol quality and the goal achievement.
Correctness metrics evaluates the adaptability and fault tolerance of the MAS regarding to the parameters change of the environment. This is performed in GAAMAAA approach through the Update Agent.
The completeness and Context coverage metrics determine the ability of the MAS to deal with the diversity of the problem domain. The use of ontologies to automatically generate the MAS makes it possible to cover the whole context (described in the ontology).

4.4 Conclusion

This chapter present a contribution to the development of a comprehensive approach encompassing the full development cycle of the MAS, from the specification of the ambient environment to the use of the MAS for providing a service. We have proposed an ontology of interaction protocols and an approach able to automatically generate a self-adaptive MAS to respond to a request for a service for an elderly or dependent person.
In particular, our contribution lies in the modeling and compilation of interaction protocols. We have proposed an algorithm capable of generating the agents from a personal assistance ontology and two algorithms (for ICNP and Coalition protocol) capable of embedding an interaction protocol in the agents. The first results of the developped prototype are very promising. The aim of the next chapter is to establish a strategy allowing an agent to dynamically detect the necessity to switch from an interaction protocol to another.

CHAPTER 5

Sensitivity analysis of GAAMAAA

5.1 Introduction

As shown in the previous chapters, interaction protocols play a crucial role in ensuring the adaptive feature in MASs. Indeed, we have shown in [Maddouri *et al.* 2019], for an ambient assisted application scenario, that interaction protocols performances can differ according to the execution context of the application. The aim of this chapter is to improve GAAMAAA by allowing to adapt the interaction protocol according to the constraints of the application problem to be solved. For that purpose, the aim is to identify the parameters that influence the performances of the protocols in order to be able to appropriately tune the protocols on the one hand, and to choose a protocol according to its performances on the other. The considered parameters, as inputs, are those related to the data of the application (ex. characteristics of the COs, intrusion level of the COs...), while the performances concern the outputs of the MAS (ex. number of solutions, time response...).
We have realized a sensitivity analysis based on the data related to the COs and the constraints of the application (level of intrusion, precision of the result). We launched requests to the MAS by varying the inputs values and observing the system outputs. The experimentation has been performed for two distinct interaction protocols: Iterative Contract Net protocol (INCP) and coalition based protocol (Coalition). The findings of the sensitivity analysis are then transformed into adaptation knowledge that will be injected into the agents before these are deployed.
Results analysis allows to build rule bases where each rule indicates a relevance context for a particular protocol. The rule bases can be used by the agents for detecting the relevance or not of a protocol and then switching from one protocol to another according their relevance. Note that the protocol changing is done at run-time thanks to GAAMAAA ability to compile an interaction protocol at run-time.

5.2 Sensitivity analysis and self-adaptive multiagent systems

Sensitivity analysis (SA) refers to the problem of sampling the parameters space when we investigate about the behaviour of a model. This term is also used to refer to a family of methods for altering the inputs of the models in different ways. In [Richiardi *et al.* 2006a] (see also [Troitzsch *et al.* 1999]), SA is defined as a collection of tools and methods used for investigating how sensitive the output values of a model are to changes in the input values. This is usually performed in the validation step of simulations ([LAW *et al.* 1991]). Noisy systems and complex systems are often validated thanks to SA methods. MASs are particularly known to be noisy due to the complexity of the interactions among the agents themselves and among the agents and the external environment. They are themselves used for SA such as in [De Santo *et al.* 2001]. In [Bhusal & Subbarao 2019] SA has been used to identify "the weakest link" during the agents communication. The approach considers different probability distribution for weighting the communications (by means of a probability density function) and then, the simulation propagates the information (agents states) between the agents. This allowed to identify the most significant links, which are those that have the highest sensitivity index compared to others.
An important challenge in our work is to identify situations where an interaction protocol is

relevant to a particular context, and the ones where it is necessary to adjust the protocol or to replace it by another that is more efficient. Thus, the idea is to empower the MAS with a self-adaptation capability. For that purpose, a SA is performed in order to reveal effects of parameter variations on the performances of the interaction protocols.
The goals of our SA are summarized in the main questions below:

1. What are the parameters that determine the performance of an interaction protocol?
2. Which areas in parameter space result in efficient protocol?
3. Which parameters have significant effects for which outputs?
4. How to generalize the results and the method of the SA in order to improve the use of the interaction protocols?

The next Section presents the sensitivity analysis approach and the parameters taken into account.

5.3 Sensitivity analysis method

SA basically consists of a statistical analysis of the effect of input variations on system outputs. In [Richiardi *et al.* 2006b] the authors identify different inputs variations which are grouped into: (I) variations of random seed and noise level, (II) variations of parameter values, (III) variations of the model, e.g. agent's decision functions, (IV) data aggregation, (V) time scale and sample size. In our case, the focus is on the second type of variation, i.e., the parameters values. In the context of our domain application, the simulation objectives are as follows:

1. Study the sensitivity of multiagent interaction protocols to the influence factors (degree of urgency, level of intrusion) and the constraints (such as the level of precision);
2. Identify the performance of each protocol in different situations. The aim is to infer the best suited protocol to deal with emergency situations while performing a task with a reduced response time and less intrusive COs;
3. Inject the inferred knowledge into the agents in the form of strategies for adapting their interaction protocols according to the situations;
4. Generalize the experimentation process in order to automatically generate a MAS with self-adaptive ability (from the AA and PrOnto ontologies).

The SA of Multiagent interaction protocols is performed in the case of an ambient assisted application. The interaction protocols which will be considered are those that have been implemented in GAAMAA. So, we will consider ICNP and Coalition protocols.
The SA has been performed for scenarios such as the one described in Chapter 3. We have identified three main parameters which are of great importance in the application context. The first parameter is the Degree of Urgency (DU) of the situation. The second parameter is the Level of

Precision (LP) of the result. The third parameter is Level of Intrusion (LI) which characterizes the COs according to their intrusiveness concerning the person (ex. a camera is more intrusive than a presence detector).
Then, we used a usage task scenario to explore the sensitivity of each of the two protocols by exhaustively exploring the parameters values. We have considered a robot localization task, since this have been studied in GAAMAA. The SA outputs are considered from the point of view of success or failure of the task on the one hand, and the response time of the system for realizing the task on the other.
In the usage scenario, the different COs, encapsulated in agents (of a MAS) are characterized in AA ontology. The range values of these characteristics are used to perform the SA. Assuming that the number of COs is known, it is then possible to launch a simulation (function executeMAS() below), for each possible combination of the COs and for each value of each characteristic, to be able to proceed to the SA processing.
Let $Sc_{Prtl} : < Agent[], LP[], DU[], LI[], t >$ be a scenario modelling, whose parameters are detailed in table 5.1. The SA experimentation consists in exhaustively considering the different

Notation	Meaning	Value scales
Prtl	Type of protocol	Coalition, ICNP
LP[]	Level of precision: corresponds to the precision of the data provided by the COs.	Scaled values: obtained by normalizing the different scales values of the different COs.
DU[]	Degree of Urgency	Corresponds to the different values of the level of urgency.
LI[]	Level of Intrusion level	Corresponds to the different values of the level of intrusion of the COs.
la	A list of agents. Each agent corresponds to a CO.	$la \in Comb(Agent[])$ where $Comb()$ is a function which calculates the combined agents sub-lists.

Table 5.1 – Scenario parameters

scenarios parameters values thanks to this algorithm 4. The function *Comb()* takes as inputs the list and the number of agents. It returns a list of sub-lists (partition in the mathematical sense) corresponding to the different combinations of the agents, by varying the sub-lists cardinality from 1 to NB. The cardinality c of the partition is:

$$c = \sum_{Ag=1}^{NB} \sum_{i=1}^{n} C_{Ag}^{i} \tag{5.1}$$

The Algorithm 3 takes as input the lists of: protocols, values of level of precision, values of degree of urgency, values of the level of intrusion and the domain task to perform. Once the MAS (composed of la, a sub-list of agents) is built with the embedded interaction protocol p, the request is sent to the MAS thanks to the function *launchQuery* with the adequate parameters values (t, du, li, lp).

Algorithm 4 SA-processing Algorithm

Input: Agent[]: list of agents, Prtl[]: list of protocols, LP[]: list of values for level of precision, DU[]: list of values for the degree of urgency, LI[]: list of values for the level of intrusion, t: a domain task, NB: number of agent.

```
Begin
la: Agent[], p: Prtl[], lp: LP[], du: DU[], li: LI[]
for p in Prtl[] do do
    for la in Comb(Agent[], NB) do do
        embeedProt(p, la)
        for du in DU[] do do
            for li in LI[] do do
                for lp in LP[] do do
                    launchQuery(t, du, li, lp)
                end for
            end for
        end for
    end for
end for
End
```

Table 5.2 shows the output variables and the variables which are used to analyze the sensitivity of the interaction protocols.

Variables	Meaning
RT	Response time.
$Prtl$	Name of the protocol: Coalition, ICNP.
Nb_Sol	Number of solutions. The MAS can provide several solutions with different parameters values for the task.
SC	Success or failure of the MAS in providing a solution. Due to the oscillation problem inherent to MAS, a time-out allows to consider that the system does not provide any solution.
lp	Level of precision.
li	Level of intrusion.
du	Degree of urgency.
la	The list of agents involved for the request.

Table 5.2 – Decision variables

5.4 Experimentation and results

This section presents the experiments carried out for investigating the computational performance of the proposed interaction protocol (ICNP and Coalition). The computational experiments have been conducted with a usage scenario such as the one described in Chapter 3. Such scenarios have been also used in [Nadia.Abchiche.Mimouni *et al.* 2013], in order to design a MAS which is able to adapt the level of precision of a task and the level of intrusion of the used

COs according to the degree of urgency of the situation. We have used the same data (OCs, tasks and interaction protocols) for carrying our experimentation. The implementation is based on the multiagent platform Jade (Agent DEvelopment Framework plateform) [Home 2004]. Also, specific APIs (Ex. Jena, `https://jena.apache.org/`) have been used for requesting the AA and PrOnto ontologies.
The system is composed of six different COs, each one encapsulated in an agent in the MAS (see Table 5.4).
The computations were done using a PC with a 2.5 GHz Intel core i5 and 8 GB RAM running under Microsoft Windows 8.
Algorithm 3 launches the MAS with parameterized localization querie. The different values of the parameters, including the level of intrusion, level of precision and the degree of urgency are summarized in Table 5.3. The characteristics of the COs, in terms of precision of the provided data and their level of intrusion are listed in Table 5.4.

Parameters	**Instantiation**
NB	Number of agents: set to six.
$Prtl$	ICNP, Coalition.
LA	Robot Agent (RA), Presence Detector Agent (APD), Fixed Camera Agent (AFC), Pan-Tilt Camera Agent (APTC), Radio Frequency Identification agent (ARFID1), Fall Detector agent (ADF0).
LP	lp is scaled in [10..100].
DU	du is scaled in $[0, 1, 2, 3]$
LI	li is scaled in $[0, 1, 2, 3]$
T	Robot localization: performed though a triangulation method.

Table 5.3 – Parameters instantiating

The goal of the localization task is to locate the robot in the habitat of a person. Indeed, we suppose that an alarm occurred and, before moving to the person, the Robot has to locate itself. This task is performed in cooperation with the AE.

LA	id	LP	LI
RA	00000	50	3
$ADF0$	00001	40	2
AFC	00002	40	1
$APTC$	00003	45	2
APD	0000 4	35	0
$ARFID1$	0000 6	20	0

Table 5.4 – Communication Objects characteritics

To assess the performance of the interaction protocols (coalition and ICNP), we have evaluated the interaction protocols separately. Each launched request has been evaluated in terms of: failure or success, the number of solutions (number of coalitions in case of Coalition protocol and constituted coordinated agents in case of INCP) and the time response of the MAS. Since the ICNP and Coalition protocols have been presented in the previous chapter, the next section will focus on presenting the results by comparing the performance of the two protocols.

5.4.1 ICNP Sensitivity analysis

We start the ICNP sensitivity analysis by measuring the response time with regard to LI, DU and LP. The Figure 5.1 shows four graphs representing the performance statistics for the experiment with a cardinality of *la* set to 5. Each color represents a specific combination of the agents as shown in the legend.
We generated four cases (a), (b), (c) and (d) corresponding to values of *li* equal respectively to 0, 1, 2 and 3. As shown, the X-axis corresponds to LP and the Y-axis to the response time.
We have chosen to vary *lp* by steps of 10. We can observe that depending of *li* the time response is different. It decreases when *li* increases and it increases with the increase of *lp*. Obviously, the time-out indicates the failure of the MAS to produce a response. We can easily infer the values of parameters for which the system fails.
As depicted in the figure, we can see that in case (a) ICNP have found solutions until the value 40 of *lp*. In cases (b) and (c) ICNP found solutions until the value 50 of *lp*. In the last case (d) ICNP found solutions until the value 60 of *lp*. To summarize, we notice that when the value of LI increases the number of solution increases. When we talk about the response time, we can observe that the optimal response time is obtained in the case (d) which is result with the highest value of *li*. Table 5.5 shows the results of the experimentation for ICNP where cardinality =5. The column labeled NBSOL shows the number of provided solutions for given values respectively for *li* and *lp*. Obviously, the MAS fails when the number of solutions is equal to zero. This occurs when *lp* value is high and *li* value is low. The figure shows the range of values of *li* and *lp* for which the MAS gives at least one solution. The involved agents are indicated in the second column, providing another parameter to indicate the context in which the MAS is able or not to provide a solution. For further details about sub list solution , see Appendix in Table A.1.

NBSOL	Agents	*li*	*lp*
2 SOLUTION	ARFID1,APD	0	10 - 20
1 SOLUTION	APD	0	30
0 SOLUTION : FAILURE	–	0	40-100
2 SOLUTION	AFC, APD	1	10- 20
2 SOLUTION	AFC, APD	1	30
1 SOLUTION	AFC	1	40
0 SOLUTION : FAILURE	–	1	50- 100
2 SOLUTION	APTC, ADF0	2	10- 40
0 SOLUTION : FAILURE	–	2	50- 100
2 SOLUTION	APTC, RA	3	10 - 40
1 SOLUTION	RA	3	50
0 SOLUTION : FAILURE	–	3	60-100

Table 5.5 – ICNP sensitivity parameters: cardinality =5

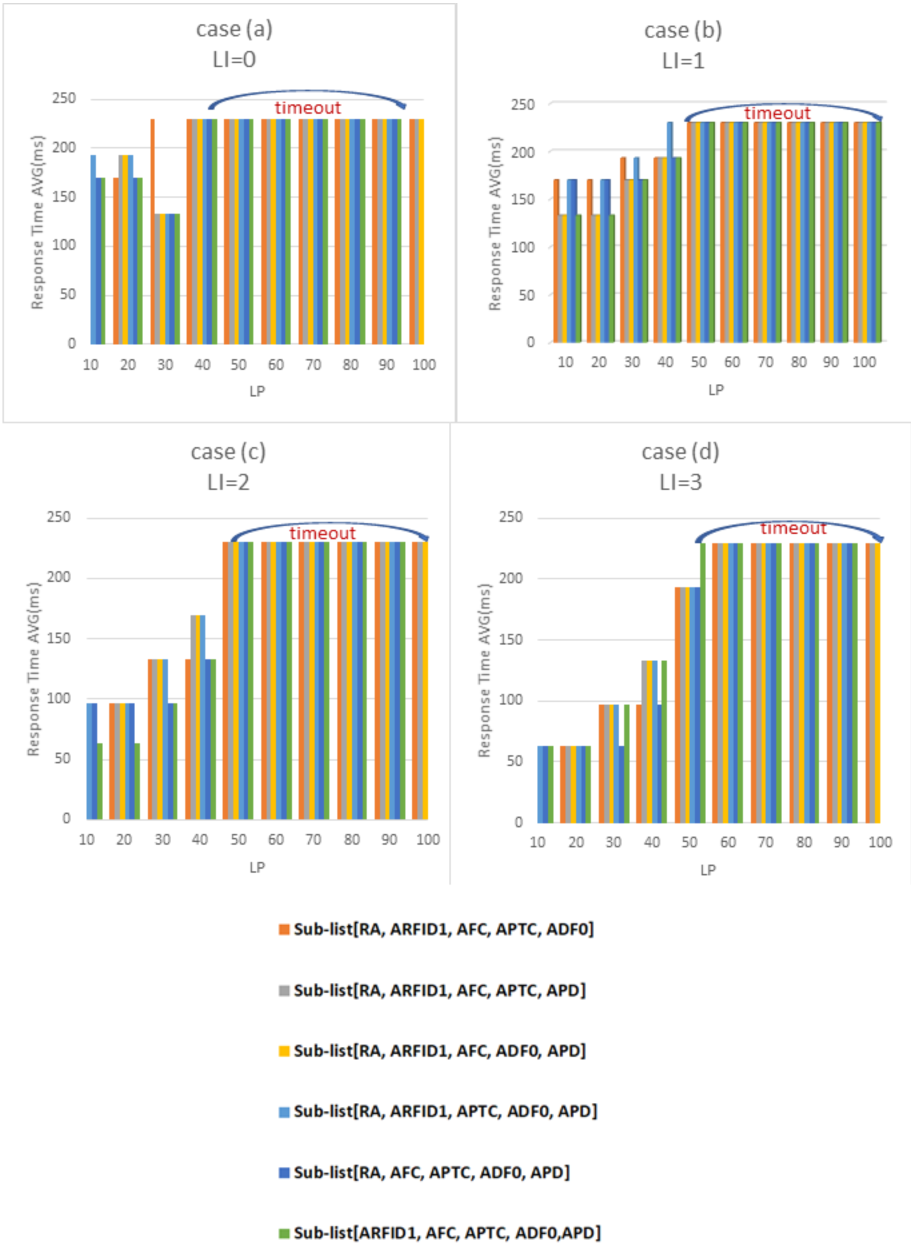

Figure 5.1 – Time response for ICNP: cardinality=5

We will show and analyze, in the result interpretation section, the other values for the cardinality 4 and 3 of the sub-lists agents.
After the sensitivity analysis of ICNP we will study, hereafter, the sensitivity analysis of a Coalition protocol.

5.4.2 Coalition-MAS Sensitivity

Launching the MAS with a coalition protocol is performed by applying the auto-protocol-coalition algorithm which creates and instantiates the MAS with a sub-list of agents. Then, the SA is performed by applying the Algorithm 3.
The results of the experimentation are showed in Figure 5.2. The four cases (a, b, c, d) show the performance coalition protocol. Each case corresponds to a particular value of *li*. The histograms represent the agents present in the system and does not prejudge the agent's membership in the coalition. Each color represents a combination of *la* as shown in the legend.
Each of the test cases corresponds to a set of agents where LP and DU vary. As depicted in the Figure (5.6), we can see that in case (a) Coalitions have been found until the value of 90 for *lp*. In other cases (b, c, d) Coalition have been found until the value 100.
To summarize, we notice that weather the value *li* increases, the number of solution and the number of coalition increase. When we talk about the response time, we can observe that the optimal response time is obtained in the first case which is result related to the number of agents which have the ability initiate a coalition (see Section 1.5.4 Classification of interaction protocols). To conclude, the influence of the *li* and the *lp* on the Coalition has an impact on the response time and the execution of the task (success of the system to provide a solution). Table depicts 5.6 that the execution of the task by coalition protocol, in case where the level of precision belongs to the interval [10, 20], the agents do not need to coalesce with other agents. Instead, each agent that has the capability to achieve the task can initiate a coalition (by sending a request to other agents). We distinguish two categories of agents:

- Agents which do not need to coalesce such as RFID1 when *lp* is in [10,20].

- Agents do not have the capacity to execute the task, need to coalesce such as DEP0 when *lp* is in [40, 100].

The table 5.6 presents the number of coalitions achieved, we have identified distinct echelon, in the high level of precision the task has performed by Coalition. For further details about sub list solution, see appendix Table A.2 and Table A.3 .

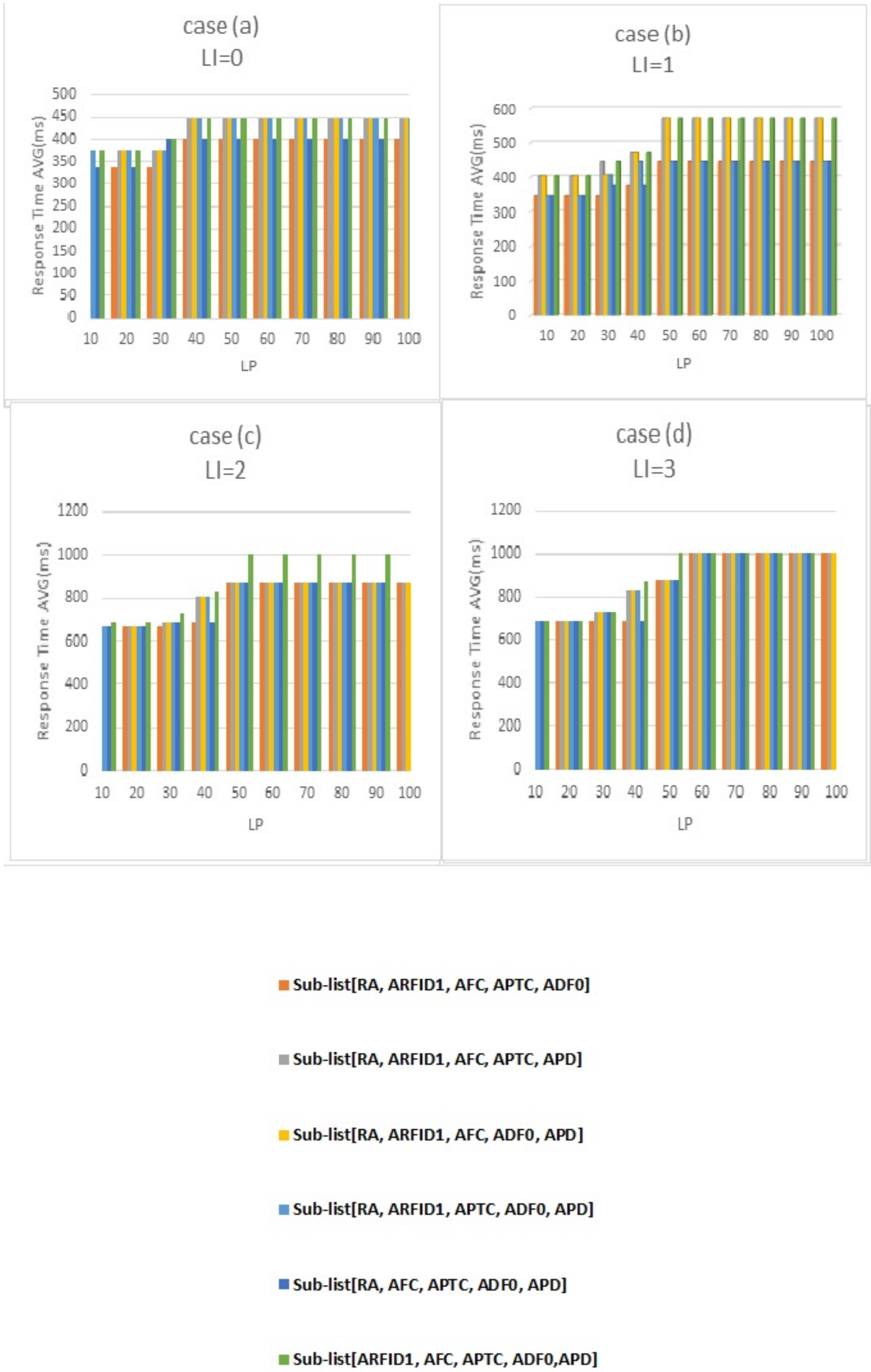

Figure 5.2 – Time response for Coalition protocol: cardinality=5

NBSOL	Agent	li	lp
2 SOLUTION	ARFID1,APD	0	10 - 20
1 SOLUTION	APD	0	30
1 SOLUTION	Coalition(ARFID1,APD)	0	40 - 50
0 SOLUTION : FAILLURE	–	0	60..100
2 SOLUTION:	AFC, APD	1	10- 30
1 SOLUTION	AFC	1	40
1 SOLUTION	Coalition(ARFID1,AFC,APD)	1	50- 90
0 SOLUTION : FAILLURE	–	1	100
2 SOLUTION	APTC, ADF0	2	10- 40
3 SOLUTION	Coalition(APTC,ADF0), Coalition(APTC, AFC), Coalition(ADF0,AFC)	2	50-80
1 SOLUTION	Coalition(AFC,APTC,ADF0)	2	90-100
2 SOLUTION	APTC, RA	3	10 - 40
2 SOLUTION	RA, Coalition(APTC,ADF0)	3	50
4 SOLUTION	Coalition(APTC,ADF0,RA), Coalition(RA,AFC,APTC), Coalition(RA,AFC,ADF0), Coalition(AFC,APTC,ADF0)	3	60- 100

Table 5.6 – Coalition sensitivity parameters: cardinality = 5

5.4.3 Synthetic results

In the last section, we studied the SA with the cardinality = 5 and checked the number of solutions in each level of intrusion and the response time. We checked using the cardinality = 3,4 and we generated the figures with four level:

- Level 1 corresponding to a sub list of the agents, each color represents a specific combination of the agents as shown in the legend.
- Level 2 corresponding to values of li equal respectively to 0, 1, 2 and 3.
- Level 3 corresponding to values of lp.
- Level 4 corresponding to results of protocol where S for succeed task and F to design failure task.

Figures 5.3 and 5.4 represent respectively the SA of protocol ICNP and coalition for cardinality = 5.
Figures 5.5, 5.7 and 5.9 show three graphs representing the SA of ICNP for the experiment with cardinality = 4.
We observed, in figure 5.7 the sensitivity of ICNP for la= [RA, AFC, APTC, ADF0], no solution was found where li=0 therefore we were obliged to increase the level of intrusion.
Figures 5.11, 5.13, 5.15 and 5.17 represent the SA of ICNP for the experiment with a cardinality = 3. In figure 5.17 the sub list [RA, ARFID1, ADP] was found to have a similar solution for li=0, 1,2.
Experimenting MAS with the coalition is shown in figures 5.6, 5.8 and 5.10 for cardinality = 4. SA coalition for cardinality = 3 is depicted in figures 5.12, 5.14, 5.16 and 5.18. We can conclude that the influence of the intrusion level and the level of precision have a impact on the

execution of task.
To study the best protocol, we need to show the cooperation between agents for this reason we don't present the results for cardinality 1,2.

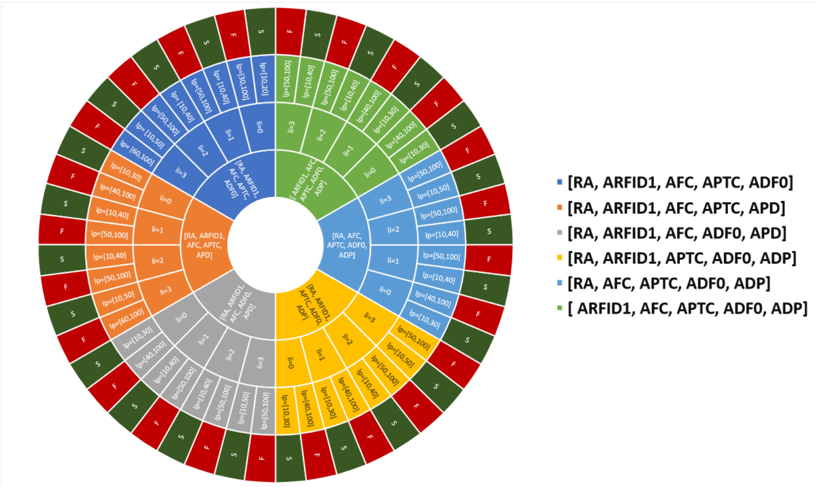

Figure 5.3 – *ICNP SA Cardinality = 5*

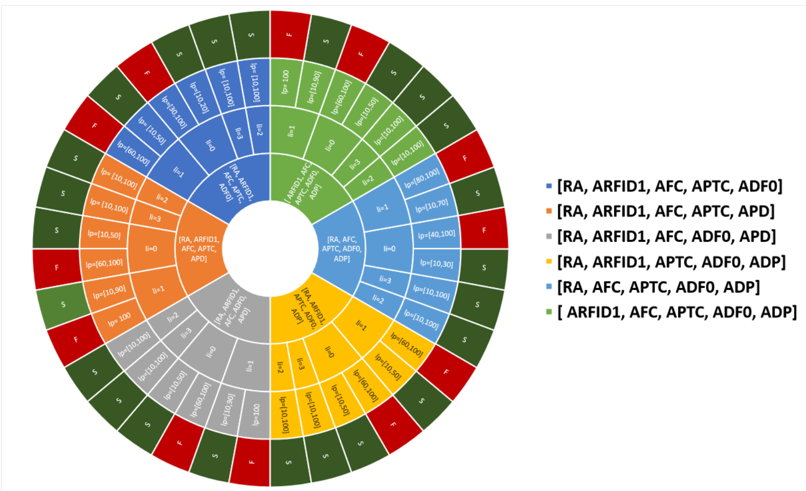

Figure 5.4 – Coalition SA Cardinality = 5

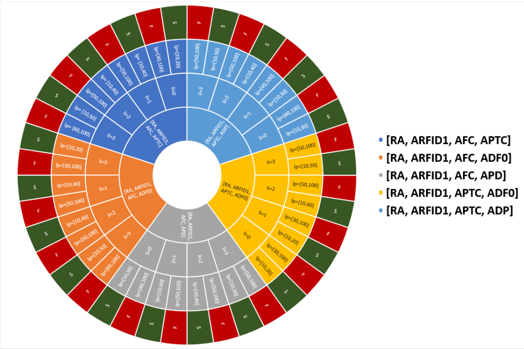

Figure 5.5 – ICNP SA Cardinality = 4

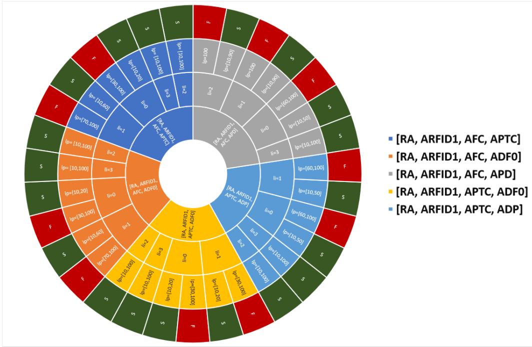

Figure 5.6 – Coalition SA Cardinality = 4

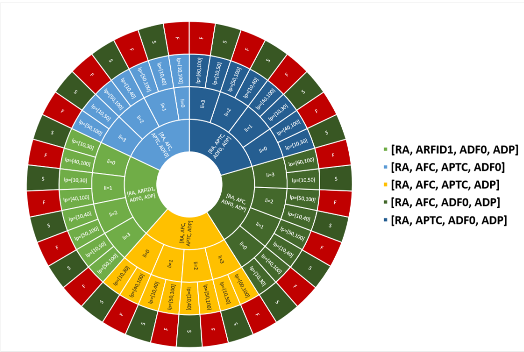

Figure 5.7 – ICNP SA Cardinality = 4

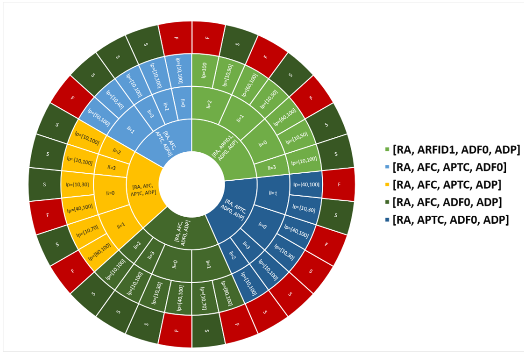

Figure 5.8 – Coalition SA Cardinality = 4

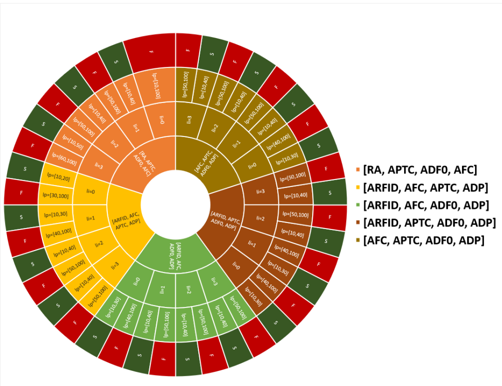

Figure 5.9 – ICNP SA Cardinality = 4

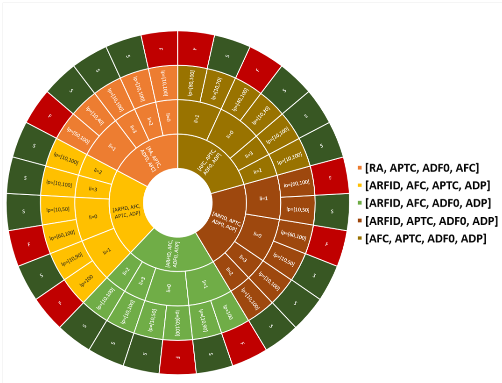

Figure 5.10 – Coalition SA Cardinality = 4

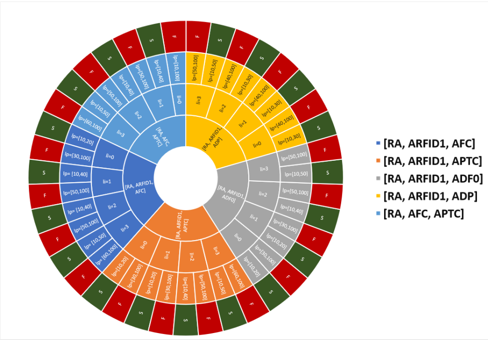

Figure 5.11 – ICNP SA Cardinality = 3

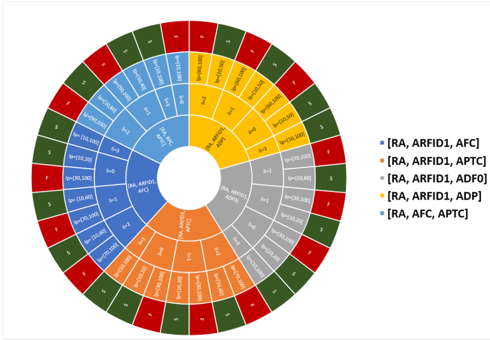

Figure 5.12 – Coalition SA Cardinality = 3

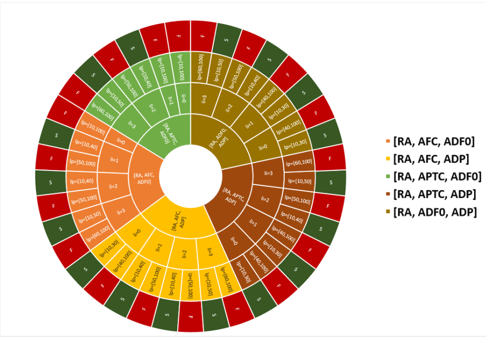

Figure 5.13 – ICNP SA Cardinality = 3

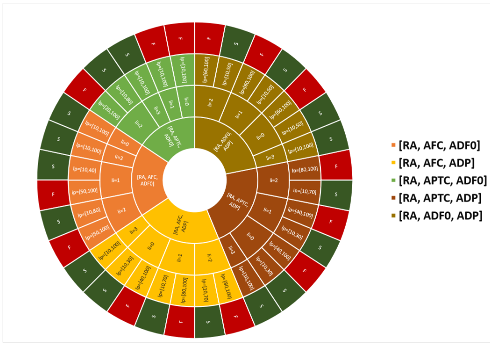

Figure 5.14 – Coalition SA Cardinality = 3

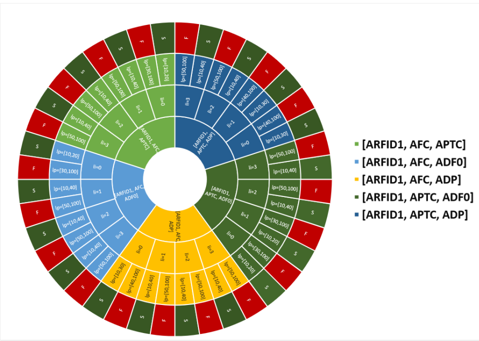

Figure 5.15 – ICNP SA Cardinality = 3

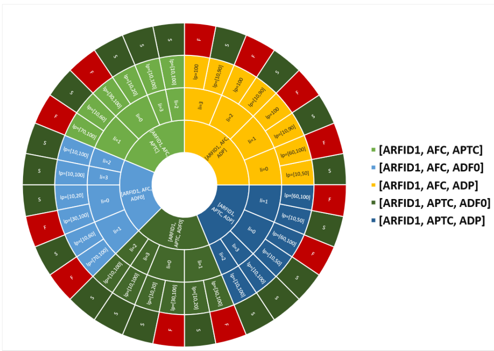

Figure 5.16 – Coalition SA Cardinality = 3

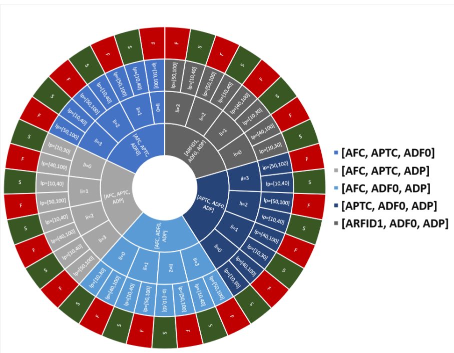

Figure 5.17 – ICNP SA Cardinality = 3

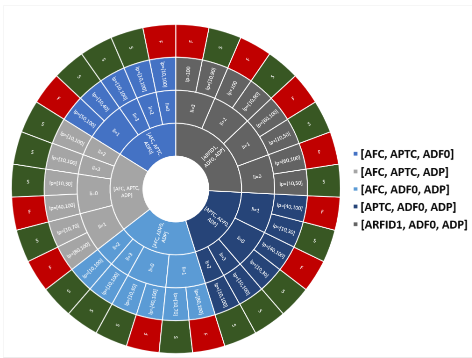

Figure 5.18 – Coalition SA Cardinality = 3

5.4.4 Result Interpretation

The results previously presented was for a cordiality that is equal to 5. For a carnality value less than 5, the results are presented in a synthetic way. One can observe the different parameters values and their impact on the success or the failure of the MAS (see Figure). Based on the experimental results reported previously, one can notice that:

- ICNP is slightly better in terms of response time but in terms of obtained results coalition protocol is better (see Figure 5.19). According to these results, we can note that in emergency situation we use ICNP if this does not provide satisfying result, we must switch to coalition protocol.

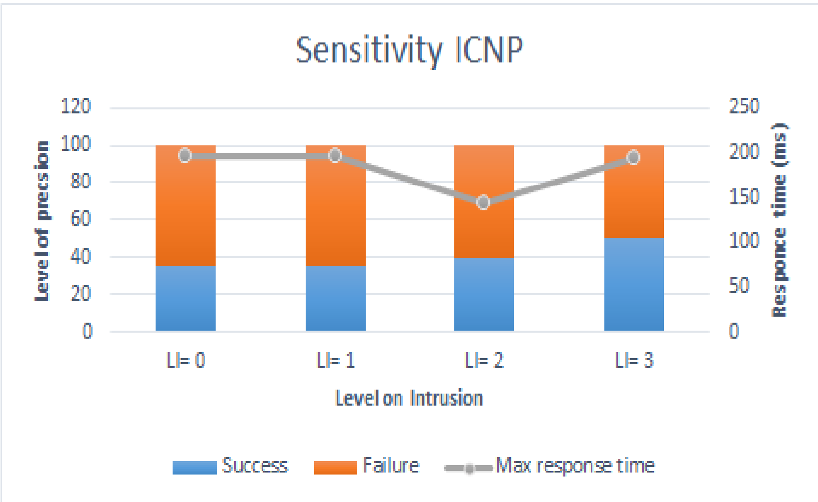

Figure 5.19 – *ICNP SA results*

- Coalition protocol allows the agents to configure themselves to provide a solution according to the availability of the COs and the compliance with the criteria of precision and level of intrusion. Consequently, the success of the task is more often ensured than with ICNP protocol.

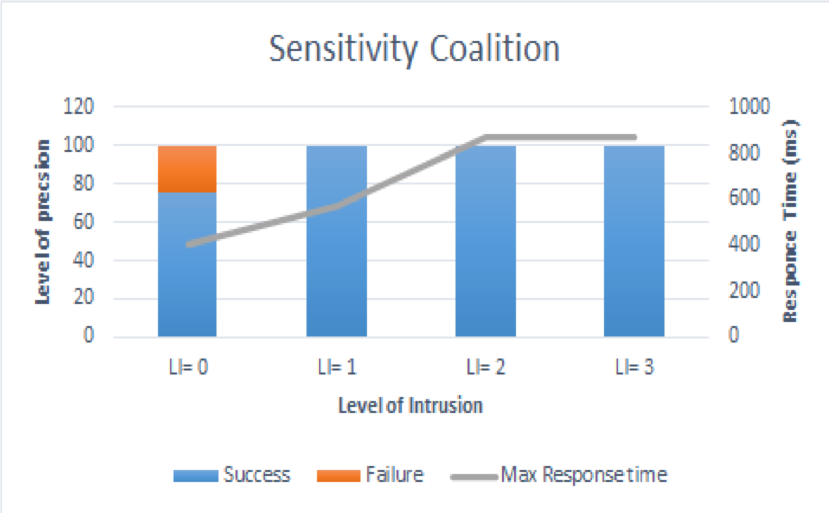

Figure 5.20 – *Coalition SA results*

We can deduce that the number of coalitions achieved is equal to the number of agents that have been solicited (for taking part to a coalition) and that have the requested level of intrusion and that do not have the sufficient level of precision. To sum up this section, it must be marked out that the ICNP and the Coalition have been tested in different situations with momentous success in terms of the quality of the obtained solutions as well as the response time. It is necessary to use the best suited protocol for each specific situation. For that purpose, the next section proposes a strategy that will allow the agents to adapt their interaction protocol accordingly.

5.5 Adaptation strategy

We have chosen to translate the contexts of the performance of the interaction protocols into production rules. These rules can then be embedded in a knowledge base which will be used by the agents to adapt their interaction protocol. The main advantage of such a representation mode is that it is easy to add, modify or delete the rules at run-time. The form of a production rule PR is: $PR : (pr_1) \wedge (pr_2) \ldots (pr_i) \ldots \wedge \ (pr_n) \Longrightarrow (Adaptation - Action)$
where: $pr_i \in premises(PR)$ $(1 \leq i \leq n)$ are the premises of the rule CR that expresses the parameters constraints). The form of the premise pr_i is defined by $pr_i = (x_i \ op \ \alpha_i)$ where x_i is the value of the i^{th} parameter, α_i is a value parameter and op is an operator comparison in (<, >, =). $Adaptation - Action$ is an action that allows a protocol adaptation. Two kinds of adaptation actions have been identified:

- Intra-protocol adaptation actions which correspond to the variation of the parameters (du, li and lp) inside the same protocol;
- Inter-protocol adaptation which corresponds to the change of protocol.

From the summary of SA of Coalition and ICNP protocols shown in Figure 5.19 and 5.20, for a cardinality set to 5, the following production rules have been defined:

- The rule $R1$ below predicts an intra protocol change for ICNP for a value of the precision level higher than 30.
  $RuleR1 : (Card = 5) \wedge (lp > 30) \wedge (Prtl = ICNP) \Longrightarrow intra - Prtl()$

- The rule $R2$ predicts an intra protocol change for ICNP for a value of the intrusion level higher than 0 and a value of the precision level less than 40.
  $RuleR2 : (Card = 5) \wedge (li > 0) \wedge (lp > 40) \wedge (Prtl = ICNP) \Longrightarrow intra - Prtl()$

- The rule $R3$ predicts an inter protocol change from ICNP to Coalition protocol for a level of intrusion equals to 1 and a value of the precision level higher than 50.
  $RuleR3 : (Card = 5) \wedge (li = 3) \wedge lp > 50 \wedge (Prtl = ICNP) \Longrightarrow Inter - Prtl()$

- The rule $R4$ predicts an intra protocol change for coalition protocol for a value of the precision level higher than 90. $RuleR4 : (Card = 5) \wedge (lp > 90) \wedge (Prtl = Coalition) \Longrightarrow intra - Prtl()$

- The rule $R5$ predicts an inter protocol change for coalition to ICNP having value of the intrusion level higher than 1 and a value of the precision level equal to 100
  $Rule5 : (Card = 5) \wedge (li > 1) \wedge (lp = 100) \wedge (Prtl = Coalition) \Longrightarrow inter - Prtl()$

The above rules have been shown to illustrate our approach. But, more rules with the different results (different values for the cardinality, LI and LP) can be defined.

5.6 Conclusion

The dynamic nature of the environment may result in ambiguous, incomplete and inconsistent contextual information, which ultimately lead the system into uncertainty. Theoretically, we cannot characterize the ambient assisted environment, so we have realized a sensitivity analysis method for determine the relevance or not of ICNP and coalition protocol based on localization scenario to determine a strategy.
Such a strategy has to be able to avoid is to run the MAS with test cases with other protocol that could induce under-performance more frequently. This supports how crucial are the type of protocol for running task. It is important to highlight that these conclusions are within the context of our test bed.
Through sensitivity analysis insight can be gained into adaptation strategy with rule-based system. The use of adaptive protocol can be predicted with the strategy. Parameters that have significant effects can be identified through rule-based system, even for complex systems. we

have establish a strategy allowing an agent to dynamically detect the necessity to switch from an interaction protocol to another. However, a method for agents has to be developed. The next step is to use an inference engine to rotate the rules and add these rules to each agent, use a trigger when certain conditions are verified or data is modified is triggered to choose the protocol and adapts to the new parameters of the rules.

Conclusion and perspectives

This chapter summarizes the thesis, we have tried to deal with the problem of the automatic generation of a cooperative and adaptive MAS applied to ambient assistive applications. Our work therefore lies at the border of several fields: multiagent systems, more specifically interaction protocols engineering and ambient assistance. An exploration of the studied context with a detailed literature review was necessary. Two main scientific contributions have been proposed (4 and 5), and raised some potential directions for future research in the studied field.

Summary of major contributions

The work presented in this document has made it possible to achieve most of the objectives we have set, namely the design and development of a framework helping to generate a self-adaptive MAS. The objective is to improve ambient assistive applications performance cope with a set of constraints, to do so, we address a localization robot scenario.
These contributions give rise to 5 chapters that are summarized as follows.

- Chapter 1: we started, by a background of aging problem, ambient assistance, MAS, interaction protocol and ontology which used to identifier the ontology of interaction protocol in chapter 4.
- Chapter 2: we presented a literature review of approaches used for multiagent systems for ambient assistance application. Then, we discussed the works. Also, we addressed a panorama of work on the engineering adaptive multiagent systems. However, there are still several shortcomings that we believe are important such as the lack of a comprehensive approach encompassing the entire development cycle of the MAS, from the specification of the ambient environment to the use of a self adaptive MAS.
- Chapter 3: We described a scenario of localization robot extracted from the Coalaa [N.Abchiche-mimouni *et al.* 2016] project. We used this scenario to evaluate GAAMAAA architecture presented in chapter 4 and to identified a strategy to make adaptation of MAS presented in chapter 5.
- Chapter 4: the main focus of this chapter is a contribution to the development of a comprehensive approach encompassing the full development cycle of the MAS, from the specification of the ambient environment to the use of the MAS for providing a service. In particular, our contribution lies in the modeling and compilation of interaction protocols. We have proposed an algorithm capable of generating the agents from a personal assistance ontology and two algorithms (for ICNP and Coalition protocol) capable of embedding an interaction protocol in the agents from the PrOnto ontology. The first results of the developed prototype are very promising.

- Chapter 5: in the previous chapter (4) engineering a MAS with a GAAMAAA architecture has provided good results without using strategy. Beyond same cases, the qualitative evaluation that are relative to the interaction protocols is not assured and the CPU time is very important. Establishing, a strategy allowing an agent to dynamically detect the necessity to switch from an interaction protocol to another is the aim of this chapter 5. The proposed strategy based on sensitivity analysis method. In order to demonstrate the performance of protocol, The simulation method for the test scenarios detailed in chapter 3 is take by the algorithm witch takes as input the list of protocol, the list of agent and the values of measure (level of precision, degree of urgency and the level of intrusion). The results are then highlighted in a graphic format. From the results of sensitivity analysis, we built rules which can be injected in the agents, allowing them to reason about the relevance of the interaction protocols.

Future works

Several improvements and perspectives are possible as a result of this work. The improvements make it possible to overcome some shortcomings of our proposals while the perspectives open new research direction. Regarding the improvements, we can note the following:

- The different proposed algorithms for engineering agents and protocols must be evaluated in terms of complexity.
- New protocols (e.g.,negotiation, vote) can be modelled in the PrOnto ontology in order to be automatically generated.
- Validation of architecture and prototypes: test this methodology on concrete examples, which will be an opportunity to obtain evaluation methodologies and apply them to real case studies, which will make it possible to highlight the contributions and limitations of our methodology.

This work also opens up many research perspectives. We can cite:

- Comparing the interaction protocols and allowing the agents ”learning” about the efficiency of the protocols and automatically generate the rules of the strategy.
- Discovery of protocols is an interesting line of research. The aim would be to answer the following questions: how, after analyzing a conversation trace, can we determine whether this trace corresponds to the execution of an existing protocol or to a new protocol? Is this really a new protocol or a new branch of an existing protocol? If it’s a new protocol how does it fit into the hierarchy of protocol types?

These different research perspectives indicate that, although research around the assessment of MAS adaptation has run out of steam in recent years, there are still some interesting and important challenges to be addressed in this area.

Appendix

I**N THIS** Appendix, we will present ambient assistance ontology (AA) and Protocol ontology (PrOnto). The two ontologies are used as one of the main techniques to build our several proposed algorithms throughout this dissertation.

Besides, it present more details of solution of ICNP and Coalition protocol.

Appendix A
Appendix 1

A screenshot showing the protocol ontology specification using protégé

A.1 Ambient assistance ontology

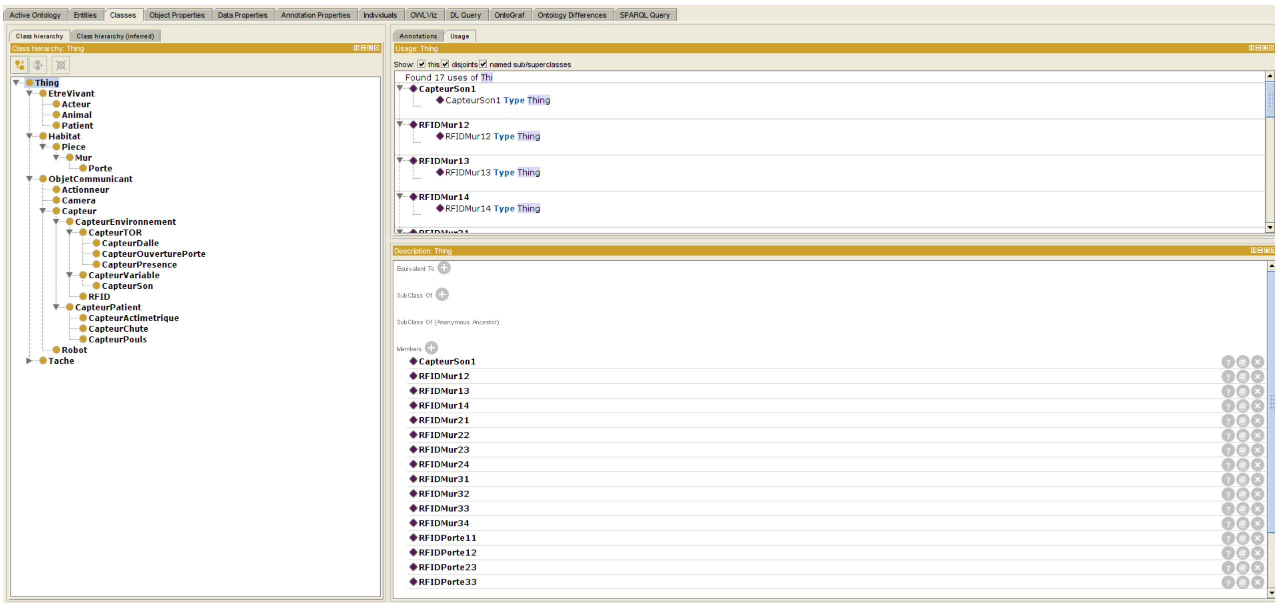

Figure A.1 – Class AA ontology

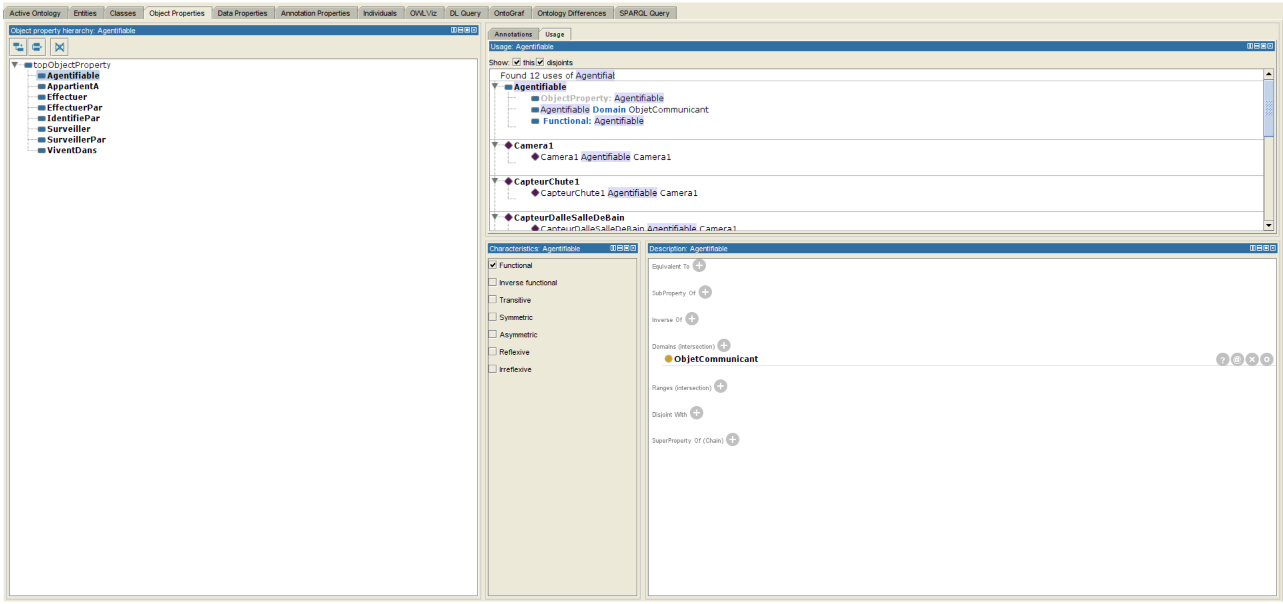

Figure A.2 – Object property AA

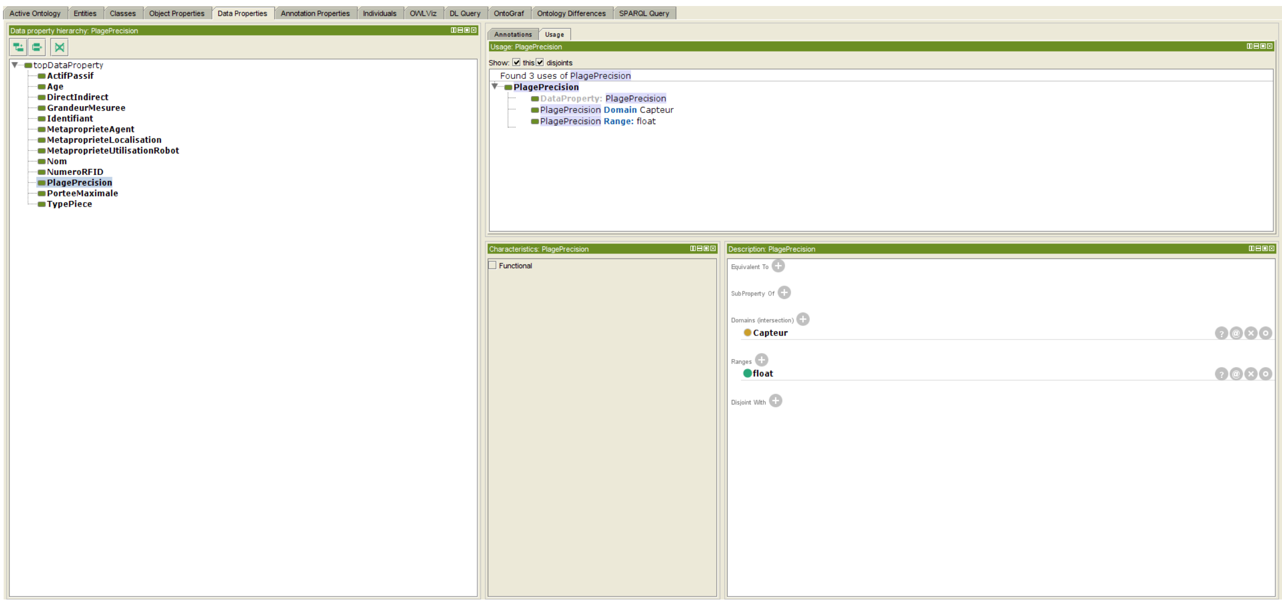

Figure A.3 – Data property AA

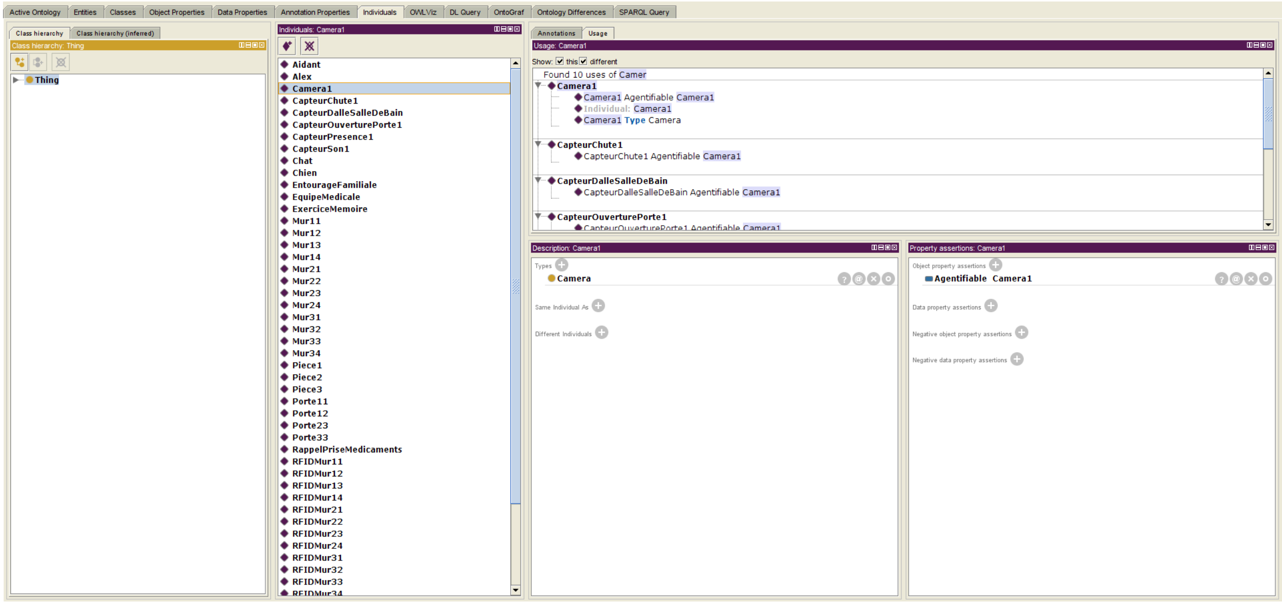

Figure A.4 – Instances AA

A.2 Protocol ontology

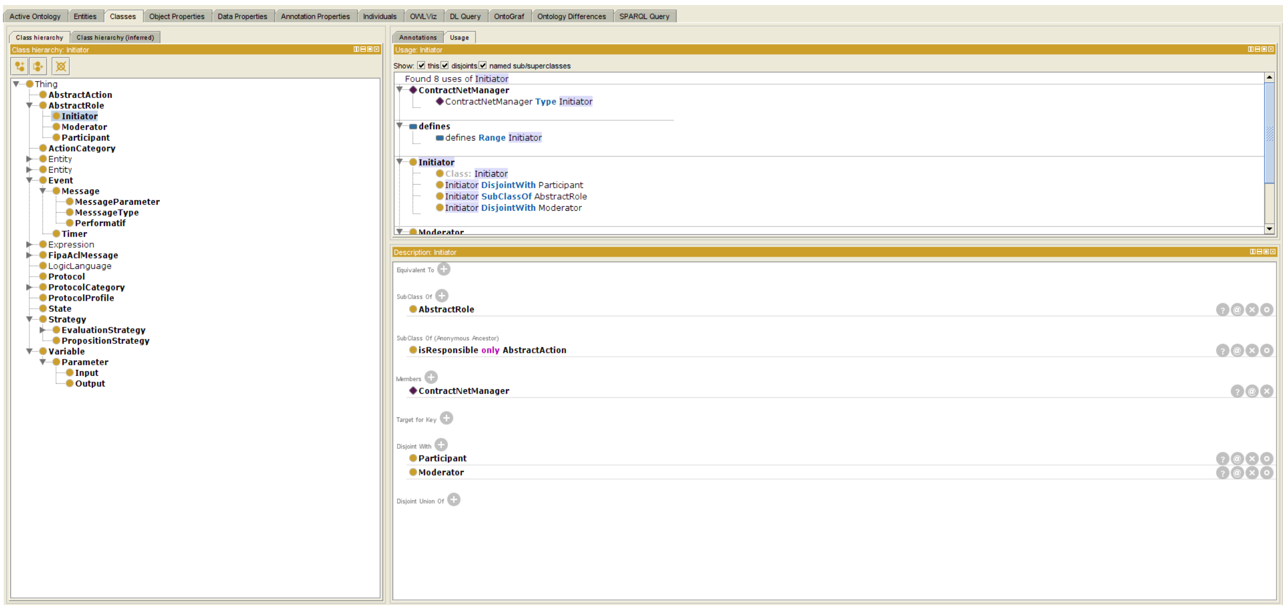

Figure A.5 – Class protocol ontology

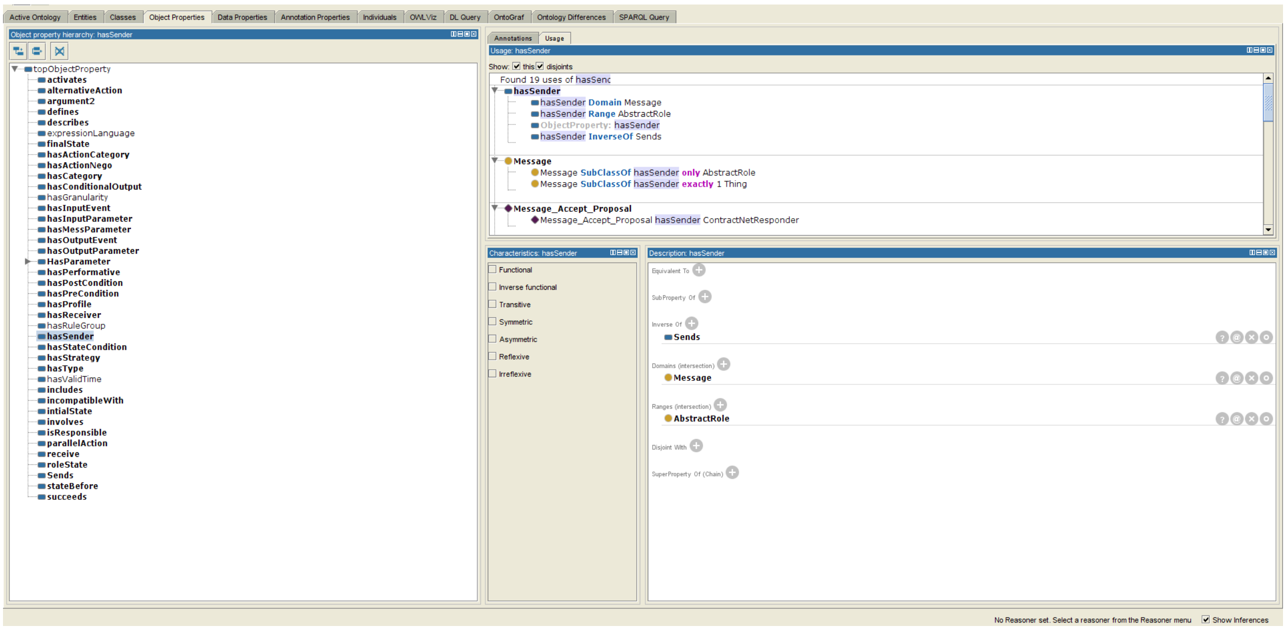

Figure A.6 – Object property

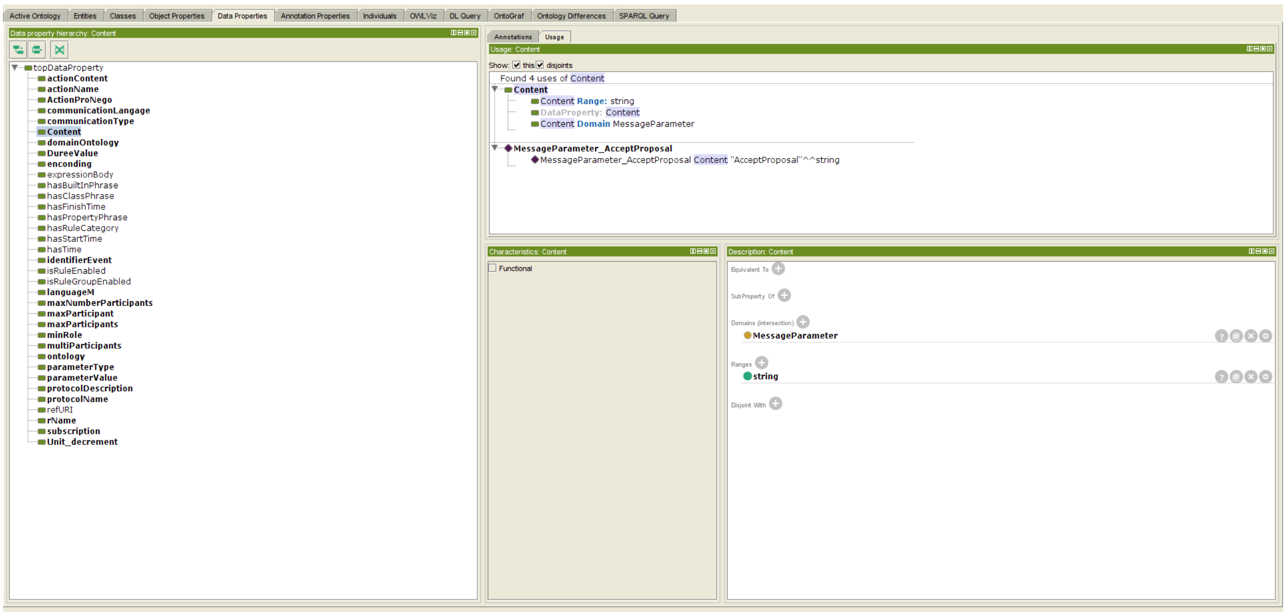

Figure A.7 – Data property

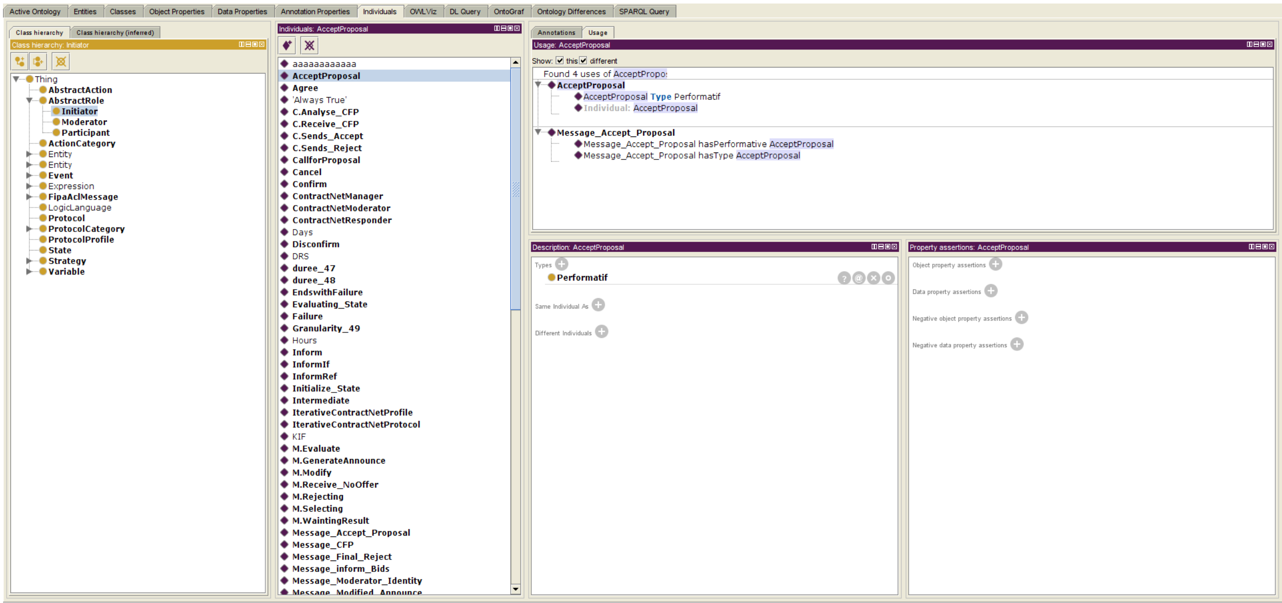

Figure A.8 – Instances

A.3 ICNP sensitivity

A tables showing the coalition protocol parameters for succeed results.

Sub-list	LI	LP	Agent
[RA, ARFID1, AFC, APTC, ADF0]	0 1 2 3	[10-20] [10-40] [10-40] [10-50]	ARFID1 AFC APTC RA
[RA, ARFID1, AFC, APTC, APD]	0 1 2 3	[10-30] [10-40] [10-40] [10-50]	APD AFC APTC RA
[RA, ARFID1, AFC, ADF0, APD]	0 1 2 3	[10-30] [10-40] [10-40] [10-50]	APD AFC ADF0 RA
[RA, ARFID1, APTC, ADF0, ADP]	0 1 2 3	[10-30] [10-30] [10-40] [10-50]	ADP ADP APTC RA
[RA, AFC, APTC, ADF0, ADP]	0 1 2 3	[10-30] [10-40] [10-40] [10-50]	ADP AFC APTC RA
[ARFID1, AFC, APTC, ADF0, ADP]	0 1 2 3	[10-30] [10-40] [10-40] [10-40]	ADP AFC APTC APTC

Table A.1 – ICNP sensitivity parameters

A.4 Coalition sensitivity

The tables showing the coalition protocol parameters for succeed results.

Sub-list (cardinality =5)	LI	LP	Agent
[RA, ARFID1, AFC, APTC, ADF0]	0	[10-20]	ARFID1
	1	[10-40]	AFC
	1	[50-60]	AFC,ARFID1
	2	[10-40]	APTC
	2	[50-90]	APTC,ADF0
	2	100	APTC,ADF0,AFC
	3	[10-50]	RA
	3	[60-90]	RA,APTC
	3	100	RA,APTC,ADF0
[RA, ARFID1, AFC, APTC, APD]	0	[10-30]	APD
	0	[40-50]	APD,ARFID1
	1	[10-40]	AFC
	1	[50-70]	AFC,APD
	1	[80-90]	AFC,APD,ARFID1
	2	[10-40]	APTC
	2	[50-80]	APTC,AFC
	2	[90-100]	APTC,AFC,APD
	3	[10-50]	RA
	3	[60-90]	RA,APTC
	3	100	RA,APTC,AFC
[RA, ARFID1, AFC, ADF0, APD]	0	[10-30]	APD
	0	[40-50]	APD,ARFID1
	1	[10-40]	AFC
	1	[50-70]	AFC,APD
	1	[80-90]	AFC,APD,ARFID1
	2	[10-40]	ADF0
	2	[50-80]	ADF0,AFC
	2	[90-100]	ADF0,AFC,APD
	3	[10-50]	RA
	3	[60-90]	RA,ADF0
	3	100	RA,ADF0,AFC

Table A.2 – Coalition sensitivity parameters (part 1)

Sub-list (cardinality =5)	LI	LP	Agent
[RA, ARFID1, APTC, ADF0, APD]	0	[10-30]	APD
	0	[40-50]	APD,ARFID1
	1	[40-50]	APD,ARFID1
	1	[10-30]	APD
	2	[10-40]	APTC
	2	[50-90]	APTC,ADF0
	2	100	APTC,ADF0,APD
	3	[10-50]	RA
	3	[60-90]	RA,APTC
	3	100	RA,APTC,ADF0
[RA, AFC, APTC, ADF0, APD]	0	[10-30]	APD
	1	[10-40]	AFC
	1	[50-70]	AFC,APD
	2	[10-40]	APTC
	2	[50-90]	APTC,ADF0
	2	100	APTC,ADF0,AFC
	3	[10-50]	RA
	3	[60-90]	RA,APTC
	3	100	RA,APTC,ADF0
[ARFID1, AFC, APTC, ADF0, APD]	0	[10-30]	APD
	0	[40-50]	APD,ARFID1
	1	[10-40]	AFC
	1	[50-70]	AFC,APD
	1	[80-90]	AFC,APD,ARFID1
	2	[10-40]	APTC
	2	[50-90]	APTC,ADF0
	2	100	APTC,ADF0,AFC
	3	[10-40]	APTC
	3	[50-90]	APTC,ADF0
	3	100	APTC,ADF0,AFC

Table A.3 – Coalition sensitivity parameters (part 2)

Bibliography

[A.Chella *et al.* 2004] A.Chella, M.Cossentino, L.Sabatucci et V.aSeidita. The PASSI and Agile PASSI MAS meta-models. Istituto di Calcolo eReti ad Alte Prestazioni, 2004. (Cited on page 22.)

[A.Eduardo *et al.* 2003] A.Eduardo, D.Kudenko et D.Kazakov. Adaptation and Multi-Agent Learning. Springer-Verlag Heidelberg, 2003. (Cited on page 51.)

[A.Ferrando *et al.* 2016] A.Ferrando, D.Ancona et V.Mascardi. Monitoring Patients with Hypoglycemia Using Self-adaptive Protocol-Driven Agents: A Case Study. International Workshop on Engineering Multi-Agent Systems, 2016. (Cited on page 33.)

[A.Freitas & R.Bordini 2017] A.Freitas et R.Bordini. V.Model-driven engineering of multi-agent. Applied Ontology, page 157–188, 2017. (Cited on page 38 and 40.)

[A.Freitas *et al.* 2017a] A.Freitas, R. H.Bordini et R.Vieira. Model-Driven Engineering of Multi-Agent Systems based on Ontologies. Applied Ontology Journal, vol. 12-2, pages 157–188, 2017. (Cited on page 9.)

[A.Freitas *et al.* 2017b] A.Freitas, R.H.Bordini et R.Vieira. Model-driven engineering of multi-agent. Applied Ontology, vol. 12, pages 157–188, 2017. (Cited on page 26.)

[A.Rao & M.Georgeff 1995] A.Rao et M.Georgeff. BDI Agents: From Theory to Practice" Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95). Menlo Park, California, AAAI Press, pages 312–319, 1995. (Cited on page 16 and 30.)

[Artur.Freitas *et al.* 2019] Artur.Freitas, Rafael H.Bordini et Renata.Vieira. Designing Multi-Agent Systems from Ontology Models. In book: Engineering Multi-Agent Systems, vol. 11375, pages 76–95, 2019. (Cited on page 9.)

[A.Tveit 2001] A.Tveit. A survey of agent-oriented software engineering. First NTNU CSGSC, 2001. (Cited on page 36.)

[Baader *et al.* 2009] Franz Baader, Ian Horrocks et Ulrike Sattler. Description Logics. Handbook on Ontologies, Staab, S. Studer, R. (Editors), Springer, page 21–43, 2009. (Not cited.)

[Bahadori *et al.* 2006] S. Bahadori, A.Cesta, L.Iocchi et G.R.Leone. Towards Ambient Intelligence For The Domestic Care Of The Elderl. Ambient Intelligence, pages 15–38, 2006. (Cited on page 30.)

[B.Chaib-Draa *et al.* 2001] B.Chaib-Draa, I.Jarras et B.Moulin. Systèmes multi-agents :principes généraux et applications. Principes et architectures des systèmes multi-agents J.-P. Briot et Y. Demazeau (Eds.)Hermès., 2001. (Cited on page 16.)

[B.Chaib-Draa 1996] B.Chaib-Draa. Interaction between Agents in Routine, Familiar and Unfamiliar; situation. International Journal of Cooperative Information Systems, vol. 5, pages 1–25, 1996. (Cited on page 16.)

[Beaumont & Chaib-draa 2007] Patrick Beaumont et Brahim Chaib-draa. Multiagent Coordination Techniques for Complex Environments : The Case of a Fleet of Combat Ships. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions, vol. 37, pages 373–385, 2007. (Cited on page 2.)

[Bechhofer *et al.* 2004] Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness et Peter F. Patel-Schneiderand Lynn Andrea Stein. OWL Web Ontology Language Reference,Technical Report. Technical Report, W3C, http://www.w3.org/TR/owl-ref/, 2004. (Cited on page 25.)

[Bhusal & Subbarao 2019] R. Bhusal et K. Subbarao. Sensitivity Analysis of Cooperating Multi-agent Systems with Uncertain Connection Weights. Dans 2019 American Control Conference (ACC), pages 4024–4029, 2019. (Cited on page 68.)

[Boissier 2001] Olivier Boissier. Modèles et architectures d'agents" Principes et architectures des systèmes multi-agent. J-P Briot et Y.Demazeau, (Eds.), vol. 1, pages 71–107, 2001. (Cited on page 15.)

[C.Bernon *et al.* 2003] C.Bernon, M-P.Gleizes, S.Peyruqueou et G.Picard. Adelfe: A methodology for adaptive multi-agent systems engineering. Engineering Societies in the Agents World III. Springer, page 156–169, 2003. (Cited on page 41.)

[Cesta *et al.* 2003] Cesta, A.Bahadori, S. G. et G.Giuliani. The robocare project, cognitive systems for the care of the elderly. In Proceedings of InternationalConference on Aging,DC, USA, 2003. (Cited on page 9.)

[Chaib-draa & Levesque 1994] Brahim Chaib-draa et Pascal Levesque. Hierarchical models and communication in multiagent environments. Proceedings of the European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds, pages 119–134, 1994. (Cited on page 16.)

[C.Mario & V.Betti 2014] C.Mario et V.Betti. Self-Adaptive Monitors for Multiparty Sessions. Euromicro International Conference on Parallel,Distributed, and Network-Based Processing, Torino, Italy, 2014. (Cited on page 41.)

[Coppo *et al.* 2014] Mario Coppo, Dezani-Ciancaglini Mariangiola et Betti Venneri. Self-Adaptive Monitors for Multiparty Sessions. Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, 2014. (Not cited.)

[D.Ancona *et al.* 2015] D.Ancona, D.Briola, A.Ferrando et V.Mascardi. Global Protocols as First Class Entities for Self-Adaptive Agents. Proceedings of the 14'th International Conference on AAMS, pages 4,8, 2015. (Cited on page 33.)

[De Santo *et al.* 2001] M. De Santo, N. Femia, M. Molinara et G. Spagnuolo. Multi agent systems for circuit tolerance and sensitivity analysis. Dans ISCAS 2001. The 2001 IEEE International Symposium on Circuits and Systems (Cat. No.01CH37196), volume 5, pages 343–346 vol. 5, 2001. (Cited on page 68.)

[D.Preuveneers & P.Novais 2012] D.Preuveneers et P.Novais. A survey of software engineering best practices for the development of smart applications in Ambient Intelligence. Journal of Ambient Intelligence and Smart Environments, vol. 4(3), page 149–162, 2012. (Cited on page 12.)

[draa Brahim *et al.* 2001] Chaib draa Brahim, Jarras Imed et Bernad Moulin. Systèmes multi-agents : principes généraux et applications. Principes et architecture des systèmes multi-agents, J. P. Briot, Y. Demazeau (dir), Paris, Hermès Science Publications, 2001. (Cited on page 14.)

[Drogoul 1993] Alexis Drogoul. De la simulation multi-agent à la résolution collective de problèmes. Une étude de l'émergence de structures d'organisation dans les systèmes multiagents. Thèse de doctorat, Université Paris 6, 1993. (Cited on page 15.)

[D.Rombach *et al.* 2011] D.Rombach, H.Storf et T.Kleinberger. Situation-of-Helplessness Detection System for Senior Citizens. Special theme: Ambient Assisted Living, pages 32–33, 2011. (Cited on page 30.)

[E.Alkadhafi *et al.* 2010] E.Alkadhafi, M.Elammari et T.Abdelaziz. Multi-Agent System Metrics. IJournal of Science and Its Applications, vol. 4, pages 52–64, 2010. (Not cited.)

[E.Kaddoum 2011] E.Kaddoum. Optimization under constraints of distributed complex problems using cooperative self-organization. URL: ftp://ftp.irit.fr/IRIT/SMAC/DOCUMENTS/ RAPPORTS/TheseElsyKaddoum.pdf., 2011. (Cited on page 42.)

[E.Michael.Bratman *et al.* 1988] E.Michael.Bratman, J.David.Israel et E.Martha.Pollack. Plans and resourcebounded practical reasoning. Computational Intelligence, vol. 4, pages 349–355, 1988. (Cited on page 16.)

[F.Aielli *et al.* 2016] F.Aielli, D.Ancona, P.Caianiello, S.Costantini, Giovanni De Gasperis, D.Giovanni, D.Marco, A.Ferrando et V. Mascardi. FRIENDLY and KIND with your Health: Human-Friendly Knowledge- INtensive Dynamic Systems for the e-Health Domain. INtensive Dynamic Systems for the e-Health Domain, Springer International publishing switzerland PAAMS, page 16–26, 2016. (Cited on page 32.)

[F.B.Hmida *et al.* 2015] F.B.Hmida, W.Lejouad et C.M.Tagina. Evaluation of Organization in Multiagent Systems for Fault Detection and Isolation. vol. 38, pages 69–79, 2015. (Cited on page 49.)

[Ferber 1999] Jacques Ferber. Multi-agent system: An introduction to distributed artificial intelligence. 1999. (Cited on page 13.)

[F.Hübner *et al.* 2007] Jomi F.Hübner, Rafael H.Bordini et Michael Wooldridge. Programming Multi-Agent Systems in AgentSpeak using Jason. John Wiley and Sons, page 273, 2007. (Cited on page 18.)

[FIPA 2002] FIPA. Fipa acl message structure specification. Technical report, 2002. (Cited on page 20.)

[FIPA 2007] FIPA. The Foundation for intelligent agents. Available: http://www.fipa.org, 2007. (Cited on page 13.)

[Foster *et al.* 2004] Ian Foster, Nicholas R.Jennings et Carl Kesselman. Brain meets brawn: Why grid and agents need each other. Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems, pages 8–15, 2004. (Cited on page 17.)

[Freitas *et al.* 2017] Artur Freitas, Rafael H.Bordini et Renata Vieira. Model-Driven Engineering of Multi-Agent Systems based on Ontologies. Applied Ontology Journal, vol. 12–2, page 157–188, 2017. (Not cited.)

[F.Zambonelli *et al.* 2003] F.Zambonelli, N.R.Jennings et M.Wooldridge. Developing multiagent systems : the gaia methodology. Software Engineering and Methodology, vol. 3, page 317–370, 2003. (Cited on page 22.)

[Gams *et al.* 2019] Matjaz Gams, Irene Y. H. Gu, Aki Härmä et Andrés Muñoz. Artificial intelligence and ambient intelligence. Journal of Ambient Intelligence and Smart Environments, vol. 11, page 71–86, 2019. (Cited on page 11.)

[Gams.Matjaz *et al.*] Gams.Matjaz, Gu.Irene Yu-Hua, H.Aki, M.Andres et Tam.Vincent. Artificial intelligence and ambient intelligence. Journal of Ambient Intelligence and Smart Environments. (Cited on page 9.)

[Georgeff *et al.* 1999] Michael Georgeff, Barney Pell, Martha Pollack, Milind Tambe et Michael Wooldridge. The belief-desireintention model of agency. In Intelligent Agents : Agents Theories, Architectures, and Languages, 1999. (Cited on page 12.)

[Giorgini *et al.* 2006] Paolo Giorgini, Manuel Kolp et John Mylopoulos. Multi-Agent Architectures as Organizational Structures. Autonomous Agents and Multi-Agent Systems, vol. 13, pages 3–25, 2006. (Cited on page 17.)

[Hallenborg *et al.* 2007] Kasper Hallenborg, Ask Just Jensen et Yves Demazeau. Reactive Agent Mechanisms for Manufacturing Process Control. IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Workshops, 2007. (Cited on page 15.)

[H.Mazouzi *et al.* 2002] H.Mazouzi, A.Saghrouchni El Fallah et S.Haddad. Open protocol design for complex interactions in multi-agent systems. AAMAS, vol. 7, page 517–526, 2002. (Cited on page 22.)

[Home 2004] JADE Home. Java Agent DEvelopment Framework (JADE). http://jade.tilab.com/, 2004. (Cited on page 39, 63 and 72.)

[IBM 2006] IBM. An architectural blueprint for autonomic computing. IBM White Paper, 2006. (Cited on page 42.)

[I.Garcia-Magarino & G.Palacios-Navarro 2016] I.Garcia-Magarino et G.Palacios-Navarro. A model-driven approach for constructing ambient assisted-living Multiagent systems customized for Parkinson patients. The Journal of Systems and Software, pages 34–48, 2016. (Cited on page 9.)

[I.Horrocks *et al.* 2004] I.Horrocks, P.Schneider, F.Peter, Boley, S.Tabet, B.Grosof et M.Dean. SWRL: A Semantic Web Rule Language. W3C Member Submission, 2004. (Cited on page 37.)

[J.Cook *et al.* 2009a] Diane J.Cook, Juan C.Augusto et Vikramaditya R.Jakkula. Ambient Intelligence: Technologies, Applications, and Opportunities. Pervasive and Mobile Computing, pages 277–298, 2009. (Cited on page 11.)

[J.Cook *et al.* 2009b] Diane J.Cook, Juan Carlos Augusto Wrede et Vikramaditya R.Jakkula. Review: Ambient intelligence: Technologies, applications, and opportunities. Pervasive and Mobile Computing, vol. 4, pages 277–298, 2009. (Cited on page 11.)

[J.Corchado *et al.* 2008] J.Corchado, J.Bajo, Y.Paz et D.Tapia. Intelligent environment for monitoring alzheimer patients. agent technology for health care, Decision Support Systems, page 382–396, 2008. (Cited on page 30 and 31.)

[J.Ferber 1995] J.Ferber. Les systèmes multi-agents. Vers une intelligence collective. Paris, 1995. (Cited on page 16.)

[J.L.Austin 1962] J.L.Austin. How to Do Things With Words. Oxford University Press, Oxford, 1962. (Cited on page 19.)

[J.M.Vidal 2008] J.M.Vidal. Multiagent Systems. Where game theory. artificial intelligence, distributed programming and the semantic web meet, page http://www.multiagent.com/bittorrentismultiagent/, March 30, 2008. (Cited on page 14.)

[J.Oukharijane *et al.* 2019] J.Oukharijane, I.B.Said, M.A.Chaabane, E.Andonoff et R.Bouaziz. Towards a New Adaptation Engine for Self-Adaptation of BPMN Processes Instancesn. ENASE 2019: Proceedings of the 14th International Conference on Evaluation of Novel Approaches to Software Engineering, 2019. (Cited on page 41.)

[J.R.Searle 1969] J.R.Searle. Speech Acts : An Essay in the Philosophy of Language. Cambridge University Press, 1969. (Cited on page 20.)

[K.Fischer *et al.* 1995a] K.Fischer, J.P.Müller et M.Pischel. Cooperative transportation scheduling: an application domain for DAI. ournal of Applied Artificial Intelligence (Special Issue on Intelligent Agents – Eds. M. J. Wooldridge and N. R. Jennings), 1995. (Cited on page 16.)

[K.Fischer *et al.* 1995b] K.Fischer, J.P.Müller et M.Pischel. Unifying control in a layered agent architecture. Proceedings of the 5th international conference on autonomous agents,Monreal- Canada, pages 324–331, 1995. (Cited on page 16.)

[Kleinberger *et al.* 2007] Thomas Kleinberger, Martin Becker, Eric Ras et Andreas Holzinger. Ambient Intelligence in Assisted Living: Enable Elderly People to Handle Future Interfaces. Ambient Interaction: International Conference on Universal Access in Human-Computer Interaction, pages 103–112, 2007. (Cited on page 12.)

[Kumar & Shapiro 1994] Deepak Kumar et Stuart C. Shapiro. The OK BDI Architecture. International Journal Of Artificial Intelligence Tools, vol. 3, pages 349–366, 1994. (Cited on page 16.)

[L.Amgoud & H.Prade 2009] L.Amgoud et H.Prade. Using arguments for making and explaining decisions 173. Artificial Intelligence, pages 413–436, 2009. (Cited on page 32.)

[LAW *et al.* 1991] LAW, A. M., KELTON et W. D. Simulation modeling and analysis. New york, McGraw-Hill, 1991. (Cited on page 68.)

[L.Damasceno *et al.* 2018] L.Damasceno, V.Werneck et M.Schots. Metric-Based Evaluation of Multiagent Systems Models. ACM/IEEE 10th International Workshop on Modelling in Software Engineering, 2018. (Cited on page 65.)

[L.Sabatucci *et al.* 2018] L.Sabatucci, V.Seidita et M.Cossentino. The Four Types of Self-adaptive Systems: A Metamodel. International Conference on Intelligent Interactive Multimedia Systems and Services, 2018. (Cited on page 41.)

[Maddouri *et al.* 2019] Amina Maddouri, Nadia Abchiche Mimouni, Samir Otmane et Jalel Akaichi. GAAMAAA: Generating Automatically an Adaptive Multiagent System for Ambient Assistive Applications. SoMeT: The 18th International Conference on Intelligent Software Methodologies, Tools, and Techniques, pages 39–54, 2019. (Cited on page 3 and 68.)

[Mahdavi-Hezavehi *et al.* 2017] S. Mahdavi-Hezavehi, P.Avgeriou et D.Weyns. A Classification Framework of Uncertainty in Architecture-Based Self-Adaptive Systems With Multiple Quality Requirements. Managing Trade-Offs in Adaptable Software Architectures, pages 45–77, 2017. (Cited on page 29 and 41.)

[M.d'Inverno *et al.* 1998] M.d'Inverno, M.Luck M.Beer, N.Jennings, C.Preist et M. Schroeder. Negotiation in Multi-Agent Systems. Workshop of the UK Special Interest Group on Multi- Agent Systems (UKMAS'98), 1998. (Cited on page 16.)

[Mekuria *et al.* 2019] Dagmawi Neway Mekuria, Paolo Sernani, Nicola Falcionelli et Aldo Franco Dragoni. A Probabilistic Multi-Agent System Architecture for Reasoning in Smart Homes. IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), pages 1–6, 2019. (Cited on page 33.)

[M.Greaves & J.Bradshaw 1999] M.Greaves et J.Bradshaw. Specifying and implementing conversation policies. Autonomous Agents Workshop, Seattle, USA, vol. Lecture Notes in Computer Science, 5132, 1999. (Cited on page 19.)

[M.Greaves *et al.* 2000] M.Greaves, H.Holmback et J.Bradshaw. What is a conversation policy ? Issues in Agent Communication, vol. 3, page 118–131, 2000. (Cited on page 22.)

[M.Hadzic *et al.* 2009] M.Hadzic, P.Wongthongtham, T.Dillon et E.Chang. Ontology-based Multi-Agent Systems. Springer, page 274, 2009. (Cited on page 25.)

[M.InvernoDavid *et al.* 1997] M.InvernoDavid, K.M.Luck et M.Wooldridge. A formal specification of dMARS. Intelligent Agents IV Agent Theories, Architectures, and Languages?Rao and Wooldridge (eds.), Lecture Notes in AI, 1365, Springer-Verlag, pages 155–176, 1997. (Cited on page 13.)

[M.J.Wooldridge & N.R.Jennings 1995] M.J.Wooldridge et N.R.Jennings. Intelligent Agents: Theory and Practice. The Knowledge Engineering Review, vol. 10(2), pages 115–152, 1995. (Cited on page 13 and 15.)

[M.Kolp *et al.* 2006] M.Kolp, P.Giorgini et J.Mylopoulos. Multi-Agent Architectures as Organizational Structures. Autonomous Agents and Multi-Agent Systems, pages 3–25, 2006. (Cited on page 17.)

[Moreno 2003] A. Moreno. Applications of software agent technology in the health care domain. Whitestein series in Software Agent Technologies, 2003. (Cited on page 31.)

[M.Reck 1994] M.Reck. Types of electronic auctions. the international conference on Information and communications technologies in tourism, 1994. (Cited on page 24.)

[N.Abchiche-mimouni *et al.* 2016] N.Abchiche-mimouni, A.Andriatrimoson, E.Colle et S.Galerne. Coalaa-gen: A general adaptive approach for ambient assistive. International Conference on Knowledge Based and Intelligent Information and Engineering, page 324–334, 2016. (Cited on page 5, 32, 33, 46, 60 and 85.)

[Nadia.Abchiche.Mimouni *et al.* 2013] Nadia.Abchiche.Mimouni, Antonio Andriatrimoson, Etienne Colle et Simon Galerne. Multidimensional adaptiveness in Multi-Agent Systems. International Journal On Advances in Intelligent Systems, IARIA, vol. 6 (1-2), pages = 124–135, 2013. (Cited on page 53, 56 and 71.)

[N.Guarino 1998] N.Guarino. Formal Ontologies and Information Systems. Amsterdam, I. P., editor, Proceedings of FOIS'98, page 3–15, 1998. (Cited on page 25.)

[Nicholas *et al.* 1998] R.Jennings Nicholas, Sycara Katia et Wooldridge Michael. A Roadmap of Agent Research and Development. Autonomous Agents and Multi-Agent Systems, vol. 1, pages 7–38, 1998. (Cited on page 13.)

[Noel 2012] Victor Noel. Component-based software architectures and multi-agent systems: Mutual and complementary contributions for supporting software development. URL: http://www.irit.fr/publis/SMAC/DOCUMENTS/RAPPORTS/TheseVictorNoel-0712.pdf, 2012. (Cited on page 42.)

[Pang & O'Neill 2019] Zhihong Pang et Zheng O'Neill. A case study of sensitivity analysis of the domestic hot water system in large Hotels. Conference Paper, January 2019. (Not cited.)

[P.Busetta & K.Ramamohanarao 1998] P.Busetta et K.Ramamohanarao. The BDIM Agent Toolkit Design. Department of Computer Sciences, Victoria. Australia, 1998. (Cited on page 16.)

[Philip.E.Agree & Chapman 1987] Philip.E.Agree et David Chapman. Pengi: An Implementation of a Theory of Activity. Proceedings of the 6th National Conference on Artificial Intelligence, San Mateo, CA: Morgan Kaufmann, pages 268–272, 1987. (Cited on page 15.)

[P.Mathieu & M-H.Verrons 2005] P.Mathieu et M-H.Verrons. The contract net protocol : GeNCA : Un modle gnral de ngociation de contrats entre agents. intelligence artificielle, pages 837–884, 2005. (Cited on page 23.)

[R.Gruber 1993] T. R.Gruber. A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition Journal, vol. 5–2, page 199–220, 1993. (Cited on page 24.)

[Richiardi *et al.* 2006a] Matteo Richiardi, Roberto Leombruni, Nicole J. Saam et Michele Sonnessa. A Common Protocol for Agent-Based Social Simulation. Journal of Artificial Societies and Social Simulation, vol. 9, no. 1, pages 1–15, 2006. (Cited on page 68.)

[Richiardi *et al.* 2006b] Matteo Richiardi, Roberto Leombruni, Nicole J. Saam et Michele Sonnessa. A Common Protocol for Agent-Based Social Simulation. Journal of Artificial Societies and Social Simulation, vol. 9, no. 1, page 15, 2006. (Cited on page 69.)

[R.Neches *et al.* 1991] R.Neches, R.Fikes, T.Finin, T.Gruber, R.Patil, T.Senator, et W. R.Swartout. Enabling technology for knowledge sharing. Enabling technology for knowledge sharing. nabling technology for knowledge sharing. AI Magazine, vol. 12(3), pages 16–36, 1991. (Cited on page 24.)

[Rodnay.A.Brooks 1986] Rodnay.A.Brooks. A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation, pages 14–23, 1986. (Cited on page 15.)

[R.Seiger *et al.* 2016] R.Seiger, S.Huber, P.Heisig et U.Assmann. Enabling self-adaptive workflow for Cyber-physical Systems. Conference on Enterprise, Business-Process and Information Systems Modeling, Ljubljana, Slovenia, page 3–17, 2016. (Cited on page 42.)

[R.Seiger *et al.* 2017a] R.Seiger, S.Huber, P.Heisig et U.Assmann. Toward a framework for selfadaptive workflows in cyber-physical systems. Springer, pages 1–18,, 2017. (Cited on page 42.)

[R.Seiger *et al.* 2017b] R.Seiger, S.Huber, P.Heisig et U.Assmann. Toward a framework for selfadaptive workflows in cyber-physical systems. Springer, pages 1–18,, 2017. (Not cited.)

[R.Smith 1980] R.Smith. The contract net protocol : high-level communication and control in a distributed problem solver. IEEE Transactions on computers, pages 1104–1113, 1980. (Cited on page 22 and 23.)

[Salehie & Tahvildari 2011] Mazeiar Salehie et Ladan Tahvildari. Towards a goal-driven approach to action selection in self-adaptive software. Software - Practice and Experience, 2011. (Cited on page 41.)

[Seidita *et al.* 2018] Valeria Seidita, Massimo Cossentino et Luca Sabatucci. The Four Types of Self-adaptive Systems: Intelligent Interactive Multimedia Systems and Services. from book Move Your Mind: Creative Dancing Humanoids as Support to STEAM Activities, pages 440–450, 2018. (Not cited.)

[Shakshuki & Reid 2015] Elhadi Shakshuki et Malcolm Reid. Multi-Agent System Applications in Healthcare: Current Technology and Future Roadmap. The 6th International Conference on Ambient Systems, Networks and Technologies, Procedia Computer Science 52, vol. 52, page 252 – 261, 2015. (Cited on page 9.)

[Sirin *et al.* 2007] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur et Yarden Katz. Pellet: A Practical OWL-DL Reasoner", Journal of Web Semantics. Journal of Web Semantics, vol. 5–2, page 51–53, 2007. (Cited on page 25.)

[S.Nwana 1996] Hyacinth S.Nwana. Software Agents: An Overview. Knowledge Engineering Review, vol. 11, pages 205–244, 1996. (Cited on page 15.)

[S.P.Ferrando & E.Onaindia 2013] S.P.Ferrando et E.Onaindia. Context aware multi-agent planning in intelligent environments. Proceedings of the Genetic and Evolutionary

Computation Conference, ACM, GECCO'2004, vol. 227, pages 22–42, 2013. (Cited on page 32.)

[T.Finin *et al.* 1994] T.Finin, R.Fritzson, D.McKay et R.McEntire. KQML as an agent communication language. Third international conference on information and knowledge management, 1994. (Cited on page 20.)

[T.Jarraya & Z.Guessoum 2007] T.Jarraya et Z.Guessoum. Reuse Interaction Protocols to Develop Interactive Agents. Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology, 2007. (Cited on page 3 and 55.)

[Tom.Mitchell *et al.* 1986] Tom.Mitchell, Jaime.Carbonell et R.Michalski. Machine Learning. National Academy of Engineering, 1986. (Cited on page 11.)

[Troitzsch *et al.* 1999] Klaus Troitzsch, Ulrich Mueller, Nigel Gilbert et James Doran. Social Science Microsimulation. J. Artificial Societies and Social Simulation, vol. 2, 01 1999. (Cited on page 68.)

[United Nations & Social Affairs 2019] Department of Economic United Nations et Population Division Social Affairs. World Population Ageing 2019. Highlights (ST/ESA/SER.A/397), 2019. (Cited on page 9.)

[V.Tamma *et al.* 2005] V.Tamma, S.Phelps, Dickinson I et M.Wooldridge. Monitoring Ontologies for supporting negotiation in e-commerce. Engineering Applications of Artificial Intelligence, page 223–236, 2005. (Cited on page 3, 38, 40 and 55.)

[W.Bouaziz & E.Andonoff 2015] W.Bouaziz et E.Andonoff. The contract net protocol : Autonomic Protocol-based Coordination in Dynamic Inter-Organizational Workflow. IEEE International Conference on Research Challenge in Information Science, 2015. (Cited on page 37 and 40.)

[W.Bouaziz *et al.* 2009] W.Bouaziz, E.Andonoff et VB.Heidelberg. Dynamic Execution of Coordination Protocols in Open and Distributed Multi-Agent Systems. KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications, 2009. (Cited on page 3, 37 and 55.)

[Wooldridge 2001] Michael Wooldridge. Intelligent Agents: The Key Concepts. Multi-Agent Systems and Applications II, vol. 2322, pages 3–43, 2001. (Cited on page 15.)

[Wooldridge 2009] Michael Wooldridge. An Introduction to MultiAgent Systems. books.google.com, 2009. (Cited on page 14.)

[Y *et al.* 2006] Okuyama F. Y, Vieira R., R. H. Bordini et Rocha Costa. An Ontology for Defining Environments within Multi-Agent Simulations. Workshop on Ontologies and Metamodeling in Software and Data Engineering, page 10, 2006. (Not cited.)

[Y.Shoham 1993] Y.Shoham. Agent-Oriented Programming. Artificial Intelligence, 1993. (Cited on page 14.)

[Yves 1995] Demazeau Yves. From interactions to collective behaviour in agent-based systems. Proceedings of the 1 st.European conference on cognitive Science. , pages 117–132, 1995. (Cited on page 16, 19 and 39.)

[Z.Pang & Z.O'Neill 2019] Z.Pang et Z.O'Neill. A case study of sensitivity analysis of the domestic hot water system in large Hotels. Conference Paper, 2019. (Cited on page 42, 48 and 49.)

[Álvaro F. Moreira *et al.* 2006a] Álvaro F. Moreira, Renata Vieira, Rafael H. Bordini Bordini et Jomi F. Hübner. Agent-Oriented Programming with Underlying Ontological Reasoning. Proceedings of the 3rd International Workshop on Declarative Agent Languages and Technologies (DALT), page 155–170, 2006. (Cited on page 26.)

[Álvaro F. Moreira *et al.* 2006b] Álvaro F. Moreira, Renata Vieira, Rafael H. Bordini Bordini et Jomi F. Hübner. Agent-Oriented Programming with Underlying Ontological Reasoning. Proceedings of the 3rd International Workshop on Declarative Agent Languages and Technologies (DALT), page 155–170, 2006. (Cited on page 26.)

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre: Contributions à l'étude et la réalisation de composants magnétiques monolithiques réalisés par PECS/SPS et à leurs applications en électronique de puissance

Mots clés: Systèmes multi-agents, Interaction, Ontologie, Objets connectés, domotique, Adaptation.

Résumé: Nos recherches portent sur les systèmes multiagents adaptatifs en environnement ouvert. Comme l'environnement est dynamique, en raison de la disponibilité ou du manque d'objets connectés et de la diversité des situations, les mécanismes de coopération entre les objets connectés doivent être conçus de manière dynamique et adaptative. L'adaptation prend en considération le contexte de la personne (personnes âgées ou malades) ainsi que les protocoles d'interaction existants.

L'objectif de ce travail est de proposer une contribution à l'automatisation de la conception et du déploiement de systèmes multi-agents adaptatifs pour les applications d'assistance ambiante. Nous proposons une nouvelle approche pour la génération automatique d'un système multi-agent adaptatif pour les applications d'assistance ambiante (GAAMAAA). Notre approche est validée par la mise en œuvre d'un scénario de localisation de robot, que le GAAMAAA prend en charge pour deux protocoles d'interaction les plus utilisés, ICNP et la coalition. L'évaluation réalisée montre la performance effective du protocole d'interaction basé sur la méthode des analyses de sensibilité.

Title: Towards auto-adaptive and cooperative multi agent systems for ambient assistive applications

Keywords: Multiagent Systems, Interaction, Ontology, Connected Objects, automation, Adaptation.

Abstract: Our research investigates adaptive Multi-Agents Systems (MAS) in open environment. Since the environment is dynamic, due to the availability or the lack of connected objects and to the diversity of situations, cooperation mechanisms among connected objects must be dynamically and adaptively designed. The adaptation takes into consideration the context of the person (elderly or sick people) as well as the existing interaction protocols.

The aim of this work is to propose a contribution to the automation of the design and the deployment of adaptive multi-agent systems for ambient assistive applications. We propose a new approach for Generating Automatically an Adaptive Multiagent system for Ambient Assistive Applications (GAAMAAA). Our approach is validated by implementing a localization robot scenario, which the GAAMAAA provides support for two most used interaction protocol ICNP and Coalition. The evaluation shows the effective performance of interaction protocol based on sensitivity analyses method.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France