The beginning is the most important part of the work.

Plato

Chapter 1

Introduction

Computer applications play an increasingly important role in everyday life. They are becoming more tightly connected with each other, forming large networks and interacting with humans through user-interfaces. Much of these systems are too complex to be completely characterized and precisely described; hence, these applications are hard to solve using centralized computing technology. Moreover, several of these systems are inherently distributed in the sense that the data and information to be processed is both geographically and temporarily distributed, or are structured into clusters whose access and use requires sophisticated capabilities [153].

We are confronted then with a new view of computing: computation as interaction, as an activity that is inherently social, and leading to new ways of conceiving, designing and developing computational systems. Agent based systems stand as a promising way to understand, manage and use these distributed, large-scale, dynamic, open and heterogeneous computing, information and social systems [85, 102]. Besides, multiagent systems offer a natural way of understanding and characterizing intelligent systems. Intelligence and interaction are deeply coupled and these systems enable us to model this insight. Several researchers argue that intelligent behaviour is not disembodied, but is a product of the interaction an agent maintains with its environment. Under this conception multiagent systems stand as a new approach to Artificial Intelligence [133].

With the spread of multiagent systems the number of projects and researchers involved in related fields has risen. Notably, there are some important coordination actions for agent based computing, as for instance AgentLink [1], an European network of researchers and developers with a common interest in agent technology. There are several websites providing information resources on intelligent agents, examples of these are Agentcities [2]

[1] http://www.agentlink.org
[2] http://grusma2.etse.urv.es/AgCitES/

and Agentland.[3]

The agent research community holds that there are some application domains where agent technologies will play a crucial role in the near future, including: ambient intelligence, grid computing, electronic business, the semantic web, bioinformatics, monitoring and control, resource management, space, military missions and manufacturing. The impact of agent technologies in application domains such as these will occur firstly as a design metaphor of complex distributed computational systems; secondly, as a source of technologies for such computing systems, and thirdly, as models of complex real-world systems [86, 102].

Considering agents as a design metaphor, they provide software designers and developers with a way of structuring an application around autonomous, communicating components, and lead to the construction of software tools and infrastructure to support this design. In order to support this view of systems development, particular tools and techniques need to be introduced. For example, methodologies that guide the analysis and design processes are required, agent architectures are needed for the design of individual software components, tools and abstractions are required to enable developers to deal with the complexity of implemented systems, and supporting infrastructure must be integrated.

In order to achieve the full potential of agent approaches and technologies there are a number of broad technological challenges for the near future. In the Agent Technology Roadmaps of the AgentLink network [86, 102], Luck et al. recommend that research and development resources should be focused along several key directions. Some of them are the following:

1. Creating tools, techniques and methodologies to support agent systems developers.
2. Automating the specification, development and management of agent systems.
3. Integrating components and features. Many different theories, architectures, technologies and infrastructures are required to specify, design, implement and manage agent based systems.
4. Establishing appropriate linkage with other branches of computer science and with other disciplines.

The work reported in this Thesis can be placed within these mentioned directions.

1.1 Motivations

This Thesis work undertakes an extension of the BDI (i.e. Belief, Desire and Intention) agent architecture in order to incorporate the representation of uncertainty in beliefs, desires —

[3] http://www.agentland.com

thus allowing the expression of graded positive and negative desires, and graded intentions. Some years ago, I worked in research projects related to knowledge-based systems, studying how different approaches to approximate reasoning could be applied to them, and turning these systems more flexible and useful for real applications [103, 100]. Now, in the framework of multiagents systems, i.e. in a distributed and complex platform of autonomous, proactive, reactive and social agents; I asked myself how the ideas underlying approximate reasoning could be extended and applied to these distributed systems.

Following this motivation, we found an interesting paper by Parsons and Giorgini [116] where a first approach to a graded BDI model was presented. It included only the representation of uncertainty in the beliefs, and left the general graded model as an open research problem. This "open door" to future work encouraged us to take this research direction. There are other contributions which treat agents that reason under uncertainty in dynamic and complex environments, but most of them deal with partial aspects of graded attitudes in intentional agents [97, 124, 142].

In this Thesis a graded BDI agent model is proposed, this model allows us to define concrete agents capable of dealing with uncertain environments (i.e. graded Beliefs) and with graded mental proactive attitudes (i.e. Desires and Intentions). Besides proposing an agent model, we consider it important to define its operational semantics to describe how a valid agent model is interpreted as sequences of computational steps. Notably, process calculi have been used to cope with formal aspects of multi-agent systems [130, 148] and we wondered if the same approach could be used to give this agent model computational meaning.

On the other hand, software engineering methodologies for developing agent based systems have become an important necessity. Even though there are valuable approaches in this field (e.g. [118, 158]), few of them emphasize the internal design of agents and consider a particular architecture. Furthermore, the actual engineering of graded BDI agents in a multiagent scenario was another relevant motivation for our work.

In the following Section, we pinpoint the contributions of this Thesis to these different fields.

1.2 Contributions

We consider that making the BDI architecture more flexible will allow us to design and develop agents capable of improved performance in uncertain and dynamic environments, serving other agents (human or not) that may have a set of graded motivations. In this research line, the central contribution of this work is the proposal of a graded BDI agent model (g-BDI), specifying an architecture capable of representing and reasoning with graded mental attitudes.

We consider this work to be an important contribution to the agent architectures field, because of the relevance of the BDI architecture and because some of our ideas may be adapted to other agent architectures. Moreover, we dealt with the operational semantics of the agent model as a first step towards a g-BDI agent interpreter and we also developed a methodology to engineer multiagent systems composed by g-BDI agents. In summary, the Thesis contributions are situated on the following diverse fields:

1. **Agent architectures:** *a general graded BDI agent model is proposed.*

   In this model, the agent graded attitudes have an explicit and suitable representation. Belief degrees represent the extent to which the agent believes a formula to be true. Degrees of positive or negative desire allow the agent to set different levels of preference or rejection respectively. Intention degrees also give a preference measure but, in this case, modelling the cost/benefit trade off of achieving an agent's goal. Then, agents having different kinds of behaviour can be modelled on the basis of the representation and interaction of their graded beliefs, desires and intentions.

   The specification of the g-BDI agent model is based on Multi-context systems (MCS). These systems were introduced by Giunchiglia et al. [68] to allow different formal (logic) components to be defined and interrelated, and Parsons et al. in [115] firstly used them to formalize BDI agents. The MCS specification of agents has several advantages pointed out by Sabater et al. in [136] both from a software engineering and a logical point of view.

   In order to represent and reason about graded notions of belief, desire and intention, in the g-BDI model we followed the approach developed by Godo et al. [72] where uncertainty reasoning is dealt with by defining suitable modal theories over suitable many-valued logics. This formalization permits us to deal with the different mental attitudes within the same well-founded logical framework.

   An illustration of the development of the g-BDI agent model and its related works is shown in Figure 1.1. The evolution of the g-BDI agent model, can be seen in [29],[30] and [31].

2. **Knowledge representation and reasoning:** *a logical framework with a sound and complete axiomatics for representing beliefs, desires and intentions is presented.*

   Looking for suitable logical systems for representing and reasoning about beliefs, desires and intentions in the g-BDI agent model is a knowledge representation problem. The question of how to deal with uncertain beliefs has been widely studied in the AI community and several approaches to approximate reasoning have been proposed (as for instance see [79]). The problem of preference representation (i.e. desires and

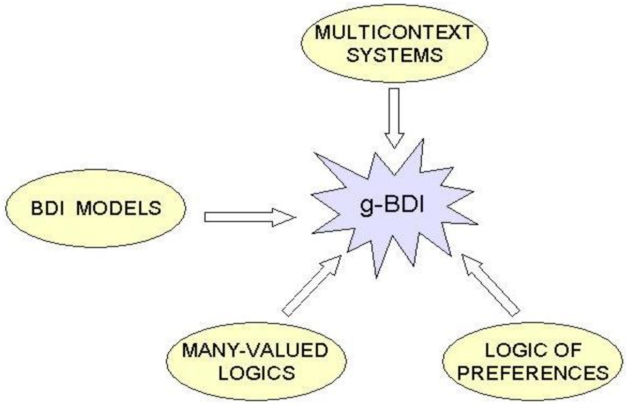

Figure 1.1: Related work to the g-BDI agent model development

intentions) has been also approached in some works (e.g. [96], [11]). Considering the desire representation in our agent model, we based our work on the bipolar model due to Benferhat et al. [12] and we extend the state of the art by giving a sound and complete axiomatics and defining different logical schemas to represent some additional constraints over preferences. In addition, we present a logical system for intentions and we show that the framework is expressive enough to describe how desires (either positive or negative), together with other information, can lead agents to intentions. Recent work in this direction was presented in [37].

3. **Process calculi:** *a Multi-context calculus (MCC) to define operational semantics for multi-context systems is developed and we use it for giving semantics to the g-BDI agent model.*

   In order to cope with the operational semantics aspects of the g-BDI agent model, we first defined a Multi-context calculus (MCC) for Multi-context systems (MCS) execution. The calculus proposed is based on Ambient calculus [28] and includes some elements of the Lightweight Coordination Calculus (LCC) [148]. We expect that MCC will be able to specify different kinds of MCSs. Particularly, we have shown how graded BDI agents can be mapped into this calculus. Through MCC we give this agent model computational meaning and in this way, we move one step closer to the development of an interpreter of the g-BDI agents. Figure 1.2 illustrates the path leading to the operational semantics of the agent model. Although process calculi have been used in the past to model multiagent systems [130, 148], we have considered that the modular structure that MCS provides to the architecture of an agent would permit a similar treatment of single agents as well. Preliminary results on the language for the execution of g-BDI agents can be seen in [36].

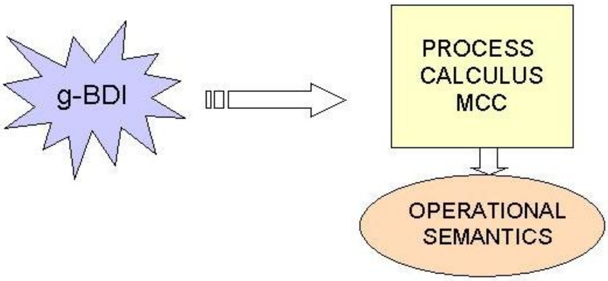

Figure 1.2: Giving Operational Semantics to the g-BDI agent model

4. **Agent based software engineering:** *a methodology for engineering agent based systems composed by agents designed as g-BDI agents, is presented.*

   We propose a software engineering process to develop graded BDI agents in a multiagent scenario. The aim of the proposed methodology is to guide the design of a multiagent system starting from a real world problem. This process is illustrated in Figure 1.3. The methodology presented has been built by adapting and extending previous approaches [88, 118, 160] in order to engineer agents with a more complex internal architecture. Furthermore, our work was inspired in some sense by the design process used in [141] where the social aspects of design are considered, and the system design phase is clearly separated from the agent design phase. Preliminary results on the methodology proposed can be seen in [34]. The design and implementation of a case study in the tourism domain is developed to show how the proposed methodology works.

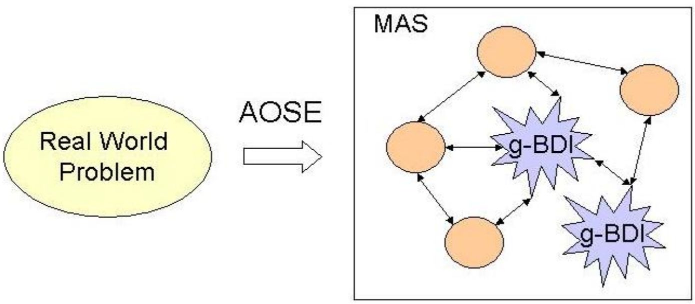

Figure 1.3: Software Engineering for the g-BDI agents

Through the design and implementation of a Tourism recommender system, where one of its principal agents is modelled as a g-BDI agent, we show in the first place that the agent model is useful to design and implement concrete agents in real world applications. The Tourism recommender design and implementation was presented in [32, 35].

Finally, using the case study we have made some experiments concerning the flexibility and performance of the g-BDI agent model. The experiments demostrate that this agent model is useful to develop concrete agents showing varied and rich behaviours. We also show that the results obtained by these particular recommender agents using graded attitudes improve those achieved by agents using non-graded attitudes. The validation and experimentation of the g-BDI Model using the case study are exposed in [38].

1.3 Structure

This Thesis is structured in four main Parts. Part I, is about Introductory Concepts and apart from the current Introduction in Chapter 2, we present related work to our research, such as: agent theories and architectures, with special attention to the BDI model; the multi-context systems and their approach to agent specification and engineering; and some logics of preference. Then, in Chapter 3 we review some of the logical background which is fundamental for our work, i.e. dynamic logic to reason about the agent actions and the transformation they produce, and some many-valued logic as Godel logic, Rational Pavelka logic and Rational Lukasiewicz logic, to reason about fuzzy modal formulae in the different contexts.

In Part II, the Graded BDI agent model is presented and then, in the consecutive Chapters 4 to 7 the general framework, its fundamental components as the different Contexts (mental and functional) and the Bridge Rules are formalized. Later on, in order to give an idea of how this model works, we show an example of a travel assistant agent. Then, the extension of the basic model that includes social and dynamic aspects is considered in Chapter 8. In Chapter 9 the operational semantics of this agent model are given.

Next, in Part III the software engineering aspects are addressed. In Chapter 10 we present the characteristics of the Tourism domain where our case study is situated. Then, in Chapter 11 a methodology to engineer g-BDI agents in a multiagent system is developed and a case study is designed using the proposed methodology. At the end of this Part, in Chapter 12, a prototype implementation of the recommender tourism system is described.

Part IV is dedicated to present our experimental work in Chapter 13 and finally, in Chapter 14 the most relevant contributions related to our Thesis work are discussed and we present some lines of future work.