PUC

ISSN 0103-9741

Monografias em Ciência da Computação
n° 13/08

Modeling Ubiquitous Applications based on Agent's Intentionality and Multi-Agent Systems

Milene Serrano

Maurício Serrano

Carlos José Pereira de Lucena

Julio Cesar Sampaio do Prado Leite

Departamento de Informática

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22451-900

RIO DE JANEIRO - BRASIL

Monografias em Ciência da Computação, No. 13/08 ISSN: 0103-9741
Editor: Prof. Carlos José Pereira de Lucena Março 2008

Modeling Ubiquitous Applications based on Agent's Intentionality and Multi-Agent Systems*

Milene Serrano[1], Maurício Serrano[1]
Carlos José Pereira de Lucena[1], Julio Cesar Sampaio do Prado Leite[1]

[1] Departamento de Informática – PUC-Rio
Rua Marques de São Vicente 225, 4º andar RDC.
Rio de Janeiro – RJ – Brasil

{milene, mauserr}@les.inf.puc-rio.br
{lucena, julio}@inf.puc-rio.br

Abstract. Current approaches for distributed computing need to be improved and extended for the ubiquitous computing domain. This new and relevant computing application area has issued original concerns and challenges, such as different devices, users and services. Thus, an appropriate approach is needed for coordinating heterogeneous computers that act as a virtual computer around a distributed, mobile and ubiquitous computing environment. Moreover, this approach has to support various services (e.g. downloads, payment, sales) that are presently beyond the capacities of single computers. In this scenario, the possibilities offered by the use of well-engineered Multi-Agent Systems (MAS) seem to offer great possibilities.

In order to investigate appropriate ways to deal with the challenges brought about by ubiquitous computing and to investigate novel software engineering methods and techniques for this domain, we have developed an exemplar application centered on a collaborative Multi-Agent System. This application provides an experimental platform for experimental research in distributed and mobile environments. For the deployment of the application, we have used several MAS concepts such as agent's intentionality and the Belief Desire Intention Model (BDI Model). Based on the described experimentation, we have developed a specific Software Engineering method for Ubiquitous Computing based on Multi-Agent Systems and Agent's Intentionality.

Keywords: Ubiquitous Computing, Software Engineering, Multi-Agent Systems, Agent Intentionality, Belief Desire Intention (BDI) Model and Goal-Oriented Planning.

Resumo. Abordagens atuais para computação distribuída precisam ser melhoradas e estendidas para o domínio da computação ubíqua. Essa nova e relevante área de aplicação da computação tem lançado desafios e preocupações originais, tais como diferentes dispositivos, usuários e serviços. Logo, uma abordagem apropriada é necessária para coordenar computadores heterogêneos que atuam como um computador virtual inserido em um ambiente computacional ubíquo, móvel e distribuído. Além disso, essa abordagem tem que suportar vários serviços (ex. downloads, pagamento e venda) que estão além das capacidades dos computadores simples. Nesse cenário, as possibilidades oferecidas pelo uso de Sistemas Multi-Agentes (SMA) parecem ser interessantes.

No intuito de investigar formas apropriadas para lidar com os desafios impostos sobre a computação ubíqua e investigar métodos e técnicas de engenharia de software inovadoras para esse domínio, foi desenvolvida uma aplicação exemplo centrada em um Sistema Multi-Agentes colaborativo. Essa aplicação provê uma plataforma experimental para pesquisas práticas em ambientes móveis e distribuídos. Para o desenvolvimento dessa aplicação, foram usados vários conceitos de SMA tais como: a intencionalidade de agentes e o Modelo *BDI* (*Belief Desire Intention*). Com base na experimentação descrita, foi desenvolvido um método de engenharia de software específico para Computação Ubíqua baseado em Sistemas Multi-Agentes e Intencionalidade de Agentes.

Palavras-Chave: Computação Ubíqua, Engenharia de Software, Sistemas Multi-Agentes, Intencionalidade de Agentes, Modelo *BDI* (*Belief Desire Intention*) e Planejamento Orientado à Meta.

* This work has been sponsored by the Ministério de Ciência e Tecnologia da Presidência da República Federativa do Brasil – CAPES.

In charge of publications

Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22451-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3527-1516 Fax: +55 21 3527-1530
E-mail: bib-di@inf.puc-rio.br
Web site: http://bib-di.inf.puc-rio.br/techreports/

Table of Contents

1 Introduction

In the last few years, researchers have been working on new technologies to provide the basic software infrastructure needed for ubiquitous computing environments [Bardram 2005] [Ballagas et al. 2003] [Yau and Karim 2001]. Most of these research works are associated with the progress being made to achieve applications that are able to deal with the inherent distributed environment, mobility of and interactions between both devices and users.

In this scenario, we particularly agree with the authors Gregory D. Abowd and Elizabeth D. Mynatt, who argue in [Abowd and Mynatt 2000]: *"The proliferation of computing into the physical world promises more than the ubiquitous availability of computing infrastructure; it suggests new paradigms of interaction inspired by constant access to information and computational capabilities."*

Based on similar ideas and the Ubiquitous Computing State-Of-The-Art [Abowd and Mynatt 2000] [Greenfield 2006] [Bell and Dourish 2007], we argue that the current approaches for distributed computing need to be improved and extended to the ubiquitous computing domain. This is due to its particular characteristics and original challenges, such as different devices, users and services. Therefore, we need an appropriate approach to coordinating heterogeneous computers that acts as a virtual computer around a distributed, mobile and ubiquitous computing environment. Moreover, this approach has to support various services (e.g. downloads, payment, sales), which are presently beyond the capacities of single computers. In this scenario, the possibilities offered by the use of well-engineered Multi-Agent Systems (MAS) seem to offer great potential.

Millions of different devices have been using the services provided by a computational infrastructure, which runs thousands of different software applications. Obviously it is impossible to manage this number of devices, services and applications without automation. In this paper, we argue that the Multi-Agent System paradigm can be used to deal with the distributed environment, user's satisfaction, different devices and several services. Moreover, it can bring a number of benefits to ubiquitous computing applications (e.g. easy ubiquitous requirement modeling and service supply). In this context, for example: (i) the agent's collaboration and reasoning techniques [Bigus and Bigus 2001] [Agotnes et al. 2007] [Pokahr et al. 2005] can be used to search, to choose and to offer the best service to a user's request; and (ii) the agent can be used as a personal assistant for each user, with capability and autonomy to learn and to adapt according to the user's preferences. This personalization has become very important for service acceptance.

Motivated by these ideas and in order to investigate the appropriate ways to deal with other new issues in ubiquitous computing, we have developed an application centered on a collaborative Multi-Agent System. We use several MAS concepts such as agent's intentionality [McBurney and Parsons 2007] [Gold and Sugden 2008] and Belief Desire Intention Model (BDI Model) [Braubach et al. 2003].

Furthermore, we describe a novel Software Engineering method for Ubiquitous Computing centered on Multi-Agent Systems and Agent's Intentionality. Subsequently, we also organize and document the State-Of-The-Art of ubiquitous computing to be able to evaluate current software engineering practices in this new context.

The remainder of this paper is organized in sections. Section 2 presents the main research issues we took into consideration. Section 3 reviews some related work. Section 4 discusses our Case Study. Section 5 describes our first attempt to propose a method that will lead to a systematic approach to ubiquitous computing. Finally, Section 6 draws some conclusions and presents our plans for future work.

2 Main Research Issues

As follows we summarize the main issues that motivated our research. They encompass the original use of existing technologies, requirements on usability, use of different user profiles and remote content servers.

- **Distributed Environment:** In ubiquitous computing, we have to consider the existence of an inherent distributed environment, which is formed by several devices and different users. Our research proposes the use of the JADE-LEAP Platform (JADE Lightweight Extensible Agent Platform) [Caire 2003] to deal with these characteristics. This platform offers some resources and services to integrate devices, even if they have different capacities. More details are presented in Section 5.
- **User's Satisfaction:** The main objective of ubiquitous computing is to provide the service to the user with quality, security, and anywhere and any time. Thus, it is necessary to consider different user's interests. Our proposal is to use a collaborative Multi-Agent System to deal with this situation. Some reasoning and learning techniques are also used to support this objective.
- **Different Devices:** Ubiquitous environments incorporate devices with different capacities and resources. In this context, our main interest is to consider the limited devices such as simple mobile phones. Our proposal investigates the use of different profiles to deal with this problem.
- **Several Services and Contents:** Our research focuses on users that request different services and content, such as music and movie downloads. Thus, we had to consider several remote content servers in our experimental ubiquitous environment.

In Figure 1, we have a typical ubiquitous environment, which illustrates the issues that interest us. We have a group of people with specific interests (e.g. music download, bank access and payment). Each person has a device. Some of these devices are powerful and others are limited. The people are in different places. This environment is distributed and requires user's satisfaction, heterogeneous devices and several services and contents.

Based on challenges such as the ones presented above, we propose to categorize some traditional Software Engineering methods and techniques, first disregarding ubiquity, while at the same time we conjecture which new Software Engineering methods and techniques are needed in the case of ubiquitous applications. Figure 2 shows the two categories of methods and tools.

Figure 1: Example of a Ubiquitous Environment

The traditional Software Engineering Approach [Pressman 2004] involves specific technologies (e.g. Object Orientation [Alencar et al. 2003], Aspect [Kiczales et al. 1999], Components [Almeida 2003], and Multi-Agent Systems [Garcia et al. 2003]), methodologies [Zambonelli et al. 2003] [Bresciani et al. 2004], platforms [Bellifemine et al. 2007] [Wallace et al. 1997], taxonomy [Buckley et al. 2005], languages [Gosling et al. 2000], CASE tools [Lending and Chervany 1998], process [Kroll and Kruchten 2003] and development environment [Boudreau et al. 2002]. Our work investigates these solutions to identify if they are appropriate to be used for ubiquitous computing. Moreover, we analyze some properties (e.g. mobility [Satoh 2005]), intelligent environments [Brumitt et al. 2000], content adaptation [Andersen et al. 2000], ontology [Harmelen and Mcguinness 2004], context aware [Baldauf et al. 2007] different devices (e.g. mobile phones, smartphones, pagers, PDAs, desktops, notebooks and others) and profiles [Machado et al. 2004] (e.g. user, network, content and device profiles).

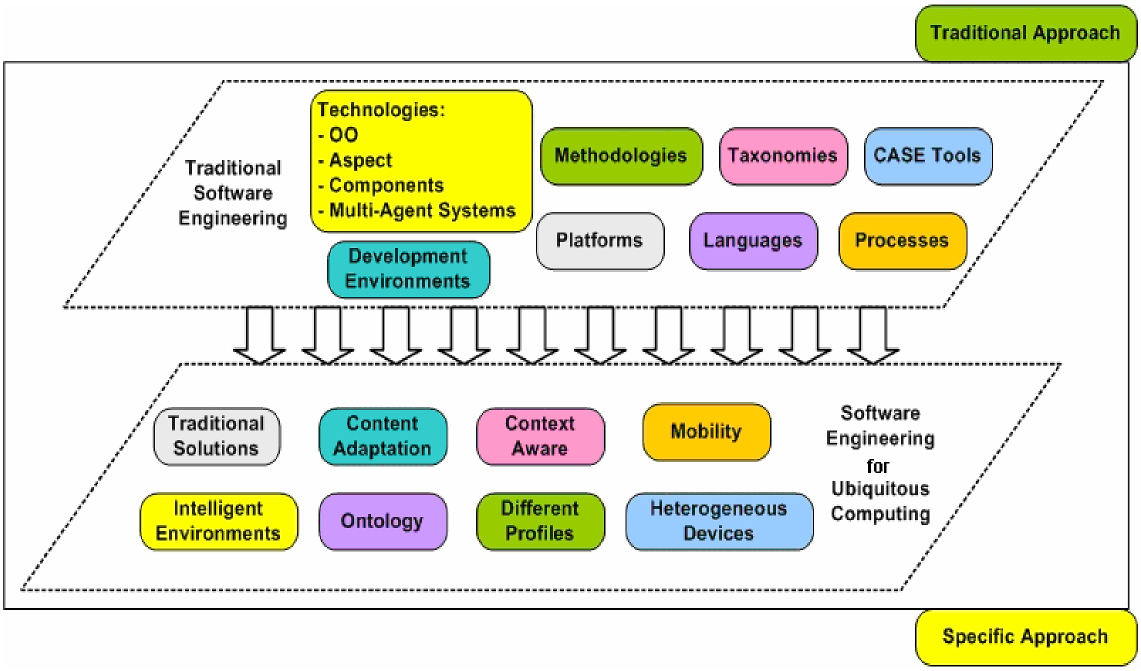

Figure 2: Traditional Software Engineering Approach and the specific Software Engineering Approach for Ubiquitous Computing

Based on our research and our experience we argue that it is possible to distinguish the traditional Software Engineering approach from a specific Software Engineering approach to ubiquitous computing. The idea is to determine which traditional and recent technologies and concepts can be combined to properly deal with ubiquitous computing issues. More clearly, the objective of the proposed categorization is to analyze the composition of the traditional and new Software Engineering solutions to reach a specific set of methods and tools in Software Engineering that meets the requirements of ubiquitous systems. Furthermore, we are particularly interested in documenting these results and developing a differentiated approach, specific to ubiquitous environments and according to a Software Engineering perspective. We think this can be done if we consider a modeling approach centered on Multi-Agent Systems and agent's intentionality.

3 Some Related Work

Several research efforts have proposed solutions to deal with Software Engineering problems, such as formalisms, languages, methodologies and environments, among others. However, only a few of them address specific problems raised by ubiquitous computing applications. Some of the work reported in the literature explores the Multi-Agent Systems paradigm. Some related work reported below bears some relationship with the ideas presented in this paper.

According to [Liogkas et al. 2004], one of the main concerns facing ubiquitous computing development is the difficulty of writing software for complex, heterogeneous and distributed applications. The authors propose an automatic application partitioning as an approach to rapid prototyping of ubiquitous computing systems. They claim this partitioning is promising for a large class of ubiquitous computing applications and present a case study using their approach. Our work has a similar purpose, which consists of investigating technologies and concepts to be used in distributed contexts. We propose an experimental environment using agents' intentionality and other State-Of-The-Art technologies (e.g. JADE-LEAP platform). Our environment can be evolved to deal with complex applications. We also argue that our methods and tools are widely applicable.

The work presented in [Ballagas et al. 2003] [O'Hare and O'Grady 2003] [Maamar et al. 2005] particularly motivated us to develop an environment based on the collaboration and interaction properties. Some of them also use intelligent agents to perform their main activities (e.g. service search and data store) and to deal with some important ubiquitous computing issues such as heterogeneous devices and users. In our case, we propose an intelligent environment centered on the BDI Model. In [Ballagas et al. 2003], *iStuff* is a toolkit of physical devices. This toolkit and a flexible software framework to support it were proposed by the authors to simplify the interaction techniques between multiple users, devices and applications. The developed environment is collaborative and interactive. The supporting software infrastructure includes a dynamically configurable intermediary for mapping of devices to applications. In [O'Hare and O'Grady 2003] *Gulliver's Genie* is a context-aware tourist guide that assists roaming tourists. The approach adopted is the deployment of intelligent agents, which collectively determine the user context and retrieve and assemble multimedia presentations. The presentations are wirelessly transmitted and displayed on a *Personal Digital Assistant (PDA)*. The research is based on beliefs, desires and intentions, which are used to determine the base deductions. Finally, the reference [Maamar et al. 2005] presents an agent-based and context-oriented approach that supports the composition process of Web services. During this process, software agents engage in conversations with their peers to agree on the Web services that will be used in this process. The authors also discuss the security of the computing resources.

Other work that is interesting and can be used on our future work is presented in [Schulzrinne et al. 2003] [Yau and Karim 2001] [Soldatos et al. 2007] [Christopoulou and Kameas 2005] [Becta 2007]. In [Schulzrinne et al. 2003], the authors describe a system based on *Session Initiation Protocol (SIP)*, with Bluetooth devices for location sensing and *Service Location Protocol (SLP)* for service discovery. They propose some solutions to ubiquitous environments, such as the introduction of context-aware location information to augment device discovery and user communication. In [Yau and Karim 2001], *Reconfigurable Context-Sensitive Middleware (RCSM)* is presented to facilitate real-time context-aware software in ubiquitous computing environments. The researchers argue that context-awareness is increasingly becoming an important capability in devices for ubiquitous computing environments. The paper [Soldatos et al. 2007] is centered on middleware for ubiquitous and context-aware computing. The authors describe several challenges, including the need for balance between transparency and context-awareness and the requirement for a certain degree of autonomy. The *GAS Ontology* [Christopoulou and Kameas 2005] is an ontology to describe the semantics of the basic concepts of a ubiquitous computing environment and to determine their inter-relations. The main goal of *GAS Ontology* is to provide a common language for the communication and collaboration among different devices that are inherent to these environments. Reference [Becta 2007] also describes emerging technologies for learning that are applicable to ubiquitous environments.

Although significant work has been undertaken in recent years, most of the research is still very application specific, with security and environmental applications. However, it is likely that a more generic and comprehensive approach is required, where different stakeholders and experts work together to solve true system level problems in the ubiquitous context. Thus, the research reported in this paper is in part motivated by the fact that there are few known specific approaches for Ubiquitous Computing from the Software Engineering viewpoint. Our work allowed us to investigate different technologies and to develop a proposal for a specific Software Engineering method for ubiquitous computing centered on Multi-Agent Systems and Agent's Intentionality.

4 Case Study Overview

We developed a Multi-Agent Systems application based on the agent's intentionality [McBurney and Parsons 2007] [Gold and Sugden 2008] and collaboration to investigate the proposed research issues and a specific Software Engineering method to support the development of Ubiquitous Computing.

Our experimental environment simulates a ubiquitous environment with different devices, users and services. Figure 3 illustrates the main idea of the case study. We consider different mobile devices in a distributed environment. Some of them are limited by memory capacity (e.g. MIDP devices[1]) and others are powerful (e.g. PJAVA devices[2]). Powerful machines assist the communication between limited mobile devices and the application to be developed. The contents and services, which are re-

1 Mobile Information Device Profile devices such as simple mobile phones.

2 Personal JAVA devices such as Smartphones.

quested by the users, are located in dedicated servers. The collaborative Multi-Agent System represents the main part of the application, which uses recently available technologies (e.g. JADE-LEAP [Caire 2003], JADEX [Pokahr and Braubach[a] 2007]; [Pokahr and Braubach[b] 2007] [Pokahr and Braubach[c] 2007], reasoning [Bigus and Bigus 2001] [Agotnes et al. 2007] [Pokahr et al. 2005] and learning [Bigus and Bigus 2001] [Lagos et al. 2007] to provide the best service based on the user's preferences (user profile). They include the device features (device profile), service/content price, server security and other parameters.

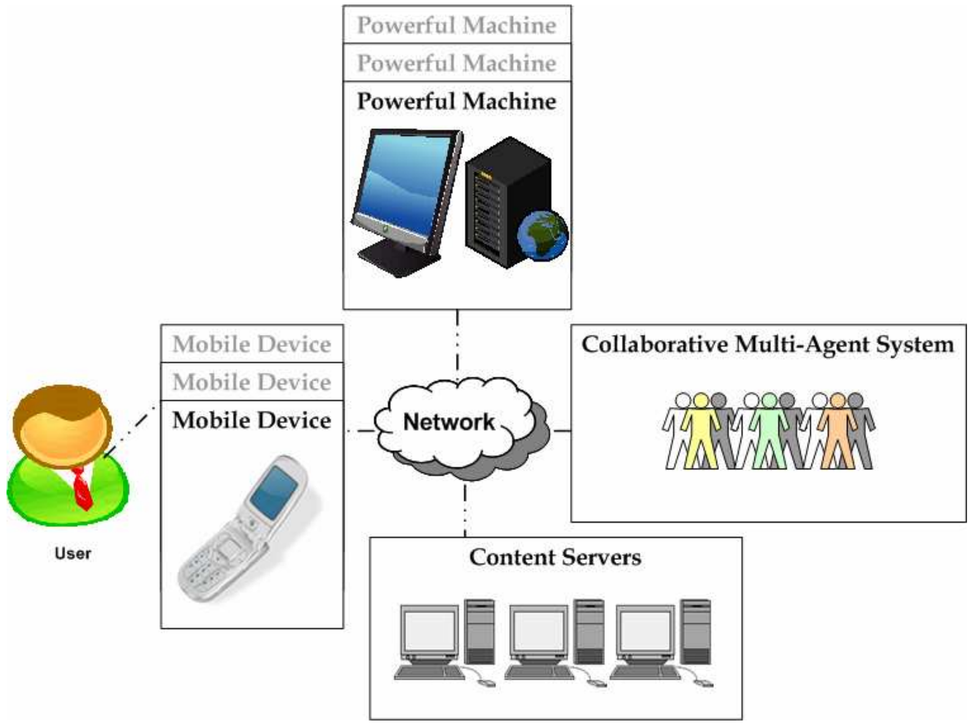

Figure 3: Experimental Environment

In order to specify the application's functionalities, we define different agents such as: *User Agent*, *Core Agent*, *Analyst Agent*, and *Music Store Agent*.

The *User Agent* is an agent that runs in the user device and communicates with *Core Agent* to request a service. That agent delegates the search, the analyses and the adaptation of the service to the Multi-Agent System, specifically to the *Core Agent*.

The *Core Agent* is the most interesting and important agent in our collaborative Multi-Agent experimental platform. It centralizes several users' requests. This agent can search and adapt the service to avoid the *User Agent* overload. It also maintains the knowledge base using learning techniques and creates an *Analyst Agent* when necessary. The *Core Agent* behaves like a cache for service requests. It verifies its knowledge base to know if it is possible to answer the service request with its own resources. If it is not possible, it creates an *Analyst Agent* to solve the problem.

The *Analyst Agent* represents the user's interests in the Multi-Agent System when the *Core Agent* delegates it. The *Analyst Agent* is created by the *Core Agent* to help it with the user's service request. It resorts to reasoning and fuzzy logic algorithms to decide which service proposal is the best among the different proposals that it receives from the *Music Store Agents*. Thus, this agent cooperates with the *Core Agent* to satisfy the user.

The *Music Store Agent* represents a mediator that contains content information, such as content location, price, security and number of services needed to access it. One or more *Music Store Agents* can interact with the *Analyst Agent* to exchange proposals and to inform the service specifications.

We have established several protocols based on FIPA Performatives and ACL Language [Braubach et al. 2003] to control the communication of the agents, which use collectivity, collaboration, autonomy, mobility and other properties to perform their activities.

The experiments implemented to validate our case study motivated our proposed approach to deal with the ubiquity issues that were first introduced in Section 2. The details of our proposed method and its applications are presented in Section 5.

5 Requirements Modeling and Ubiquitous Environment Integration Based on Cognitive Multi-Agent Systems

Our first research results allow us to propose ways (i) to model the ubiquitous requirements; (ii) to integrate a distributed ubiquitous environment; (iii) to use a collaborative ubiquitous environment based on the Multi-Agent Systems Paradigm; and (iv) to build an intelligent ubiquitous environment. The ideas are described in sections 5.1 to 5.4.

5.1 Ubiquitous Requirements Modeling centered on the Agent's Intentionality

In order to model the ubiquitous requirements, we propose the use of the i* [Yu 1998] framework that models the agents based on their intentionality. The i* supports the functional and non-functional requirements. The advantages of using this framework, which were observed during our research, are: **(1)** the reduction of the complexity of the commonly used diagrams [Bauer and Odell 2005] [Silva and Lucena 2003] [Silva and Lucena 2004]; **(2)** the modeling of functional (e.g. goals, tasks, subgoals and subtasks) [Dubois 1989] and non-functional (e.g. softgoals) requirements centered in the Goal-Orientation [Cunha 2007] that enhances the possibility of implementing traceability policies.

i* uses two diagrams to model the application requirements: (i) the *Strategic Dependency (SD) Model*, which describes the dependency relationships among actors/actors, actors/agents and agents/agents in an organizational context, and (ii) the *Strategic Rationale (SR) Model*, which describes the stakeholder interests, and how they might be addressed by different software specifications.

An example of a SD model based on our case study, explained on Section 4, is presented in Figure 4. The *User Actor* (Client) depends on the *User Agent* to find the service. It does not matter how the User Agent will achieve this state. Thus it is a Goal Dependency. The *User Agent* depends on the *User Actor* to send the user and service information. Objects of the real world, physical or informational, are modeled as resources. Thus it is a Resource Dependency. This model represents the shared goal ("Service Be Found"), the resource ("User And Service Information") and the dependencies relationships between the *User Actor* and the *User Agent*.

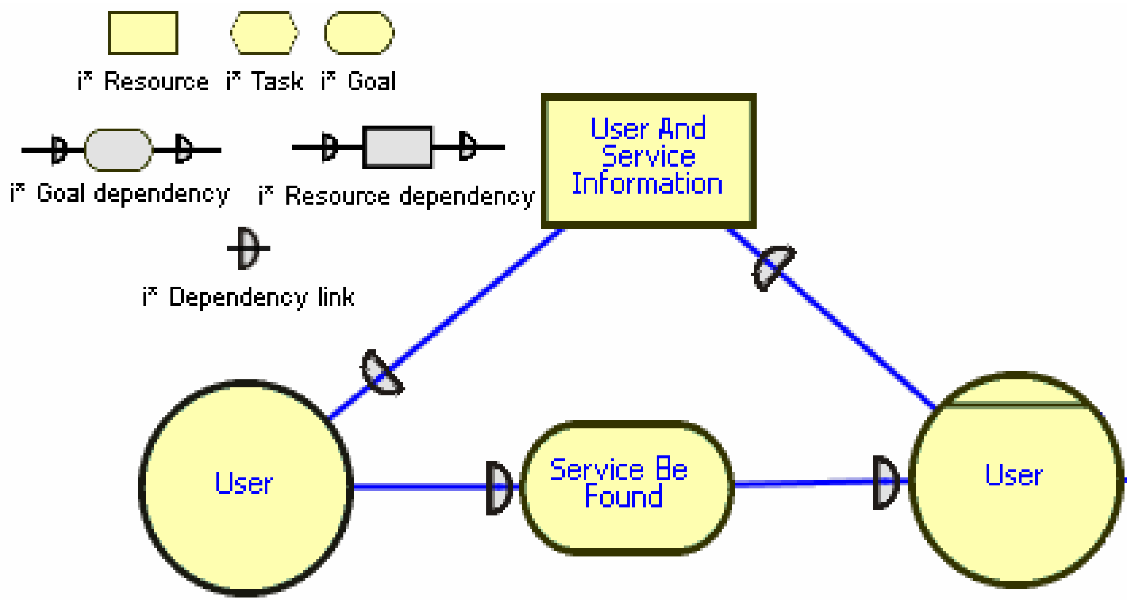

Figure 4: Actor-Agent Strategic Dependency Model

Figure 5 refers to the *User Agent-Core Agent Strategic Dependency Model*. The *User Agent* depends on the *Core Agent* to choose the service. It is necessary to send the user and service information. This model represents the goal ("Service Be Chosen"), the resources ("User Information" And "Service Information") and the dependency relationships between the *User Agent* and the *Core Agent*.

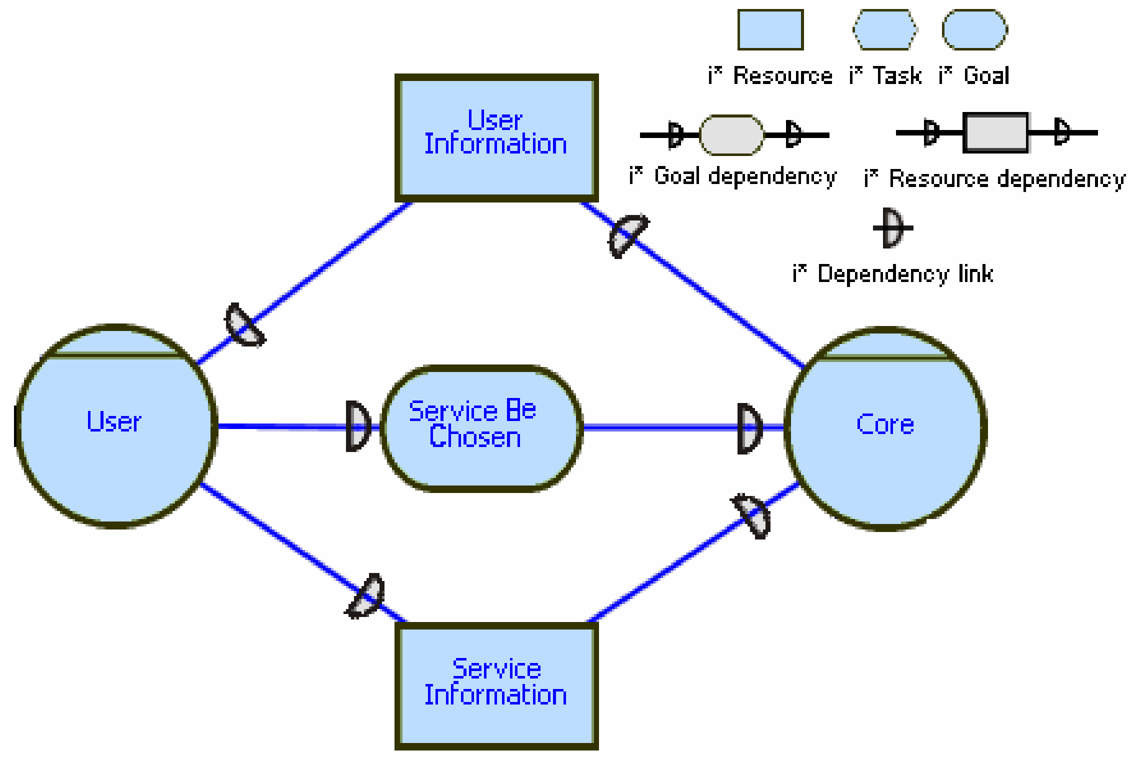

Figure 5: Agent-Agent Strategic Dependency Model

In a Task Dependency, the *depender*[3] agent imposes how to perform a task on the *dependee*[4] agent. To obtain agents with more autonomy, the task dependencies among agents must be analyzed. They should be replaced, if it is possible, by goal dependencies. A Goal Dependency must replace a Task Dependency if it is not necessary to perform the dependency in a specific way. For example, if the *dependee* agent has to find a service but it is not important what resources it will use to do this, we have to model this dependency ("Service Be Chosen") as a Goal Dependency. Otherwise, if it has to use a specific *script*, we have to model this dependency ("Service Be Chosen") as a Task Dependency. In our case, we chose to analyze all agents' dependencies. Wherever appropriate, we used Goal Dependency. Thus, the *dependee* agent becomes more autonomous as it can decide how to achieve the goal.

The i* SR Model describes the strategic agent's rationale. The goals, subgoals, tasks, subtasks, resources, means-end reasoning, tasks decomposition and softgoals are modeled for each system actor and agent. The alternatives and their contributions to softgoals can be observed by analyzing the some (+) contribution and the some (–) contribution. Figure 6 illustrates part of the *Core Agent Strategic Rationale Model*.

The *Core Agent* exchanges information with: (i) the *User Agent* to receive the user service request and the resources "User Information" and "Service Information"; (ii) the *DF Agent* to be registered and deregistered. It is necessary to send the resource "Core Agent Description" for *DF*; (iii) the *AMS Agent* to create the *Analyst Agent*. It is necessary to send the resource "Analyst Agent Definition File" for *AMS*; and (iv) the *Analyst Agent* to delegate the user request service. It is necessary to send the resources "User Information" and "Service Information" for *Analyst*.

We illustrate some non-functional requirements using the softgoals: "Performance(MAS)", "Precision(Service)", "Avoid Overload(MAS)" and "Avoid Overload(Core Agent)". The "some +" representation helps the associated softgoal to be *satisficed*[5]. The "some -" representation hurts the associated softgoal. i* forces the mapping of the alternatives and their contributions to softgoals. Thus, design decisions can be traced back to the requirements by future consultations and analyses. It is a way to maintain the traceability between requirements and code. This practice aims at evolutionary maintenance (e.g. change management and change impact analysis).

3 The depending agent is called *depender* [Yu 1995].

4 The agent who is depended upon is called *dependee* [Yu 1995].

5 In our context, the term "satisficed" is quite different than "satisfied". It means "satisfied to a sufficient degree". This term was commonly used by the Requirement Engineering researchers [Mylopoulos et al. 1992].

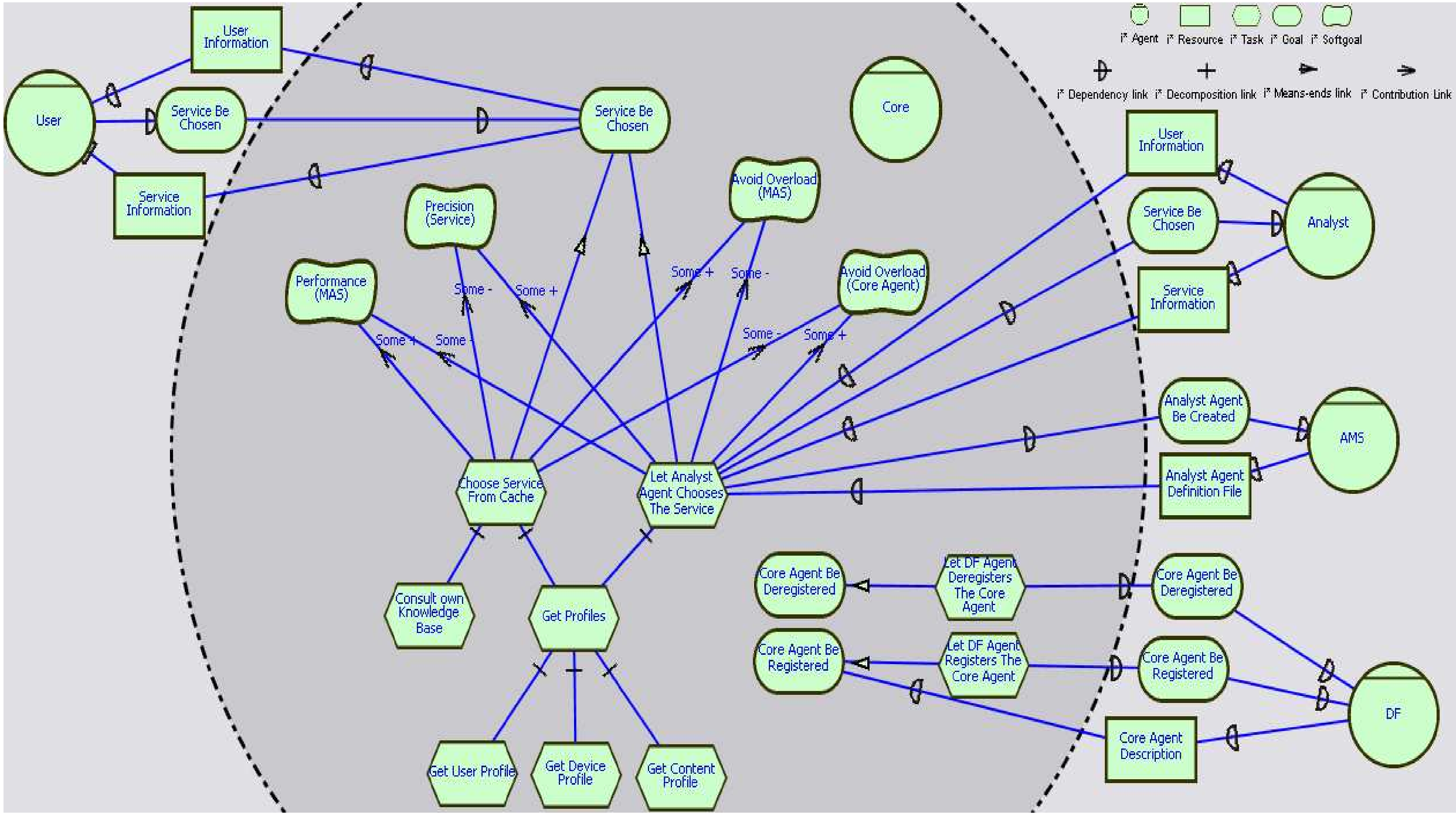

Figure 6: Core Agent Strategic Rationale Model

5.2 A Distributed Ubiquitous Environment Integration using the JADE-LEAP Platform

We considered different mobile devices in our research. Some of them are limited in their capacities and have specific features such as MIDP devices. Others are powerful and have different features such as PJAVA devices. In order to deal with these kinds of devices, we use the JADE-LEAP[6] Platform. This platform supports two execution modes: *split* and *stand-alone*. The *split* mode is appropriate to MIDP devices and the *stand-alone* mode is adequate to PJAVA devices. Figure 7 shows the execution modes.

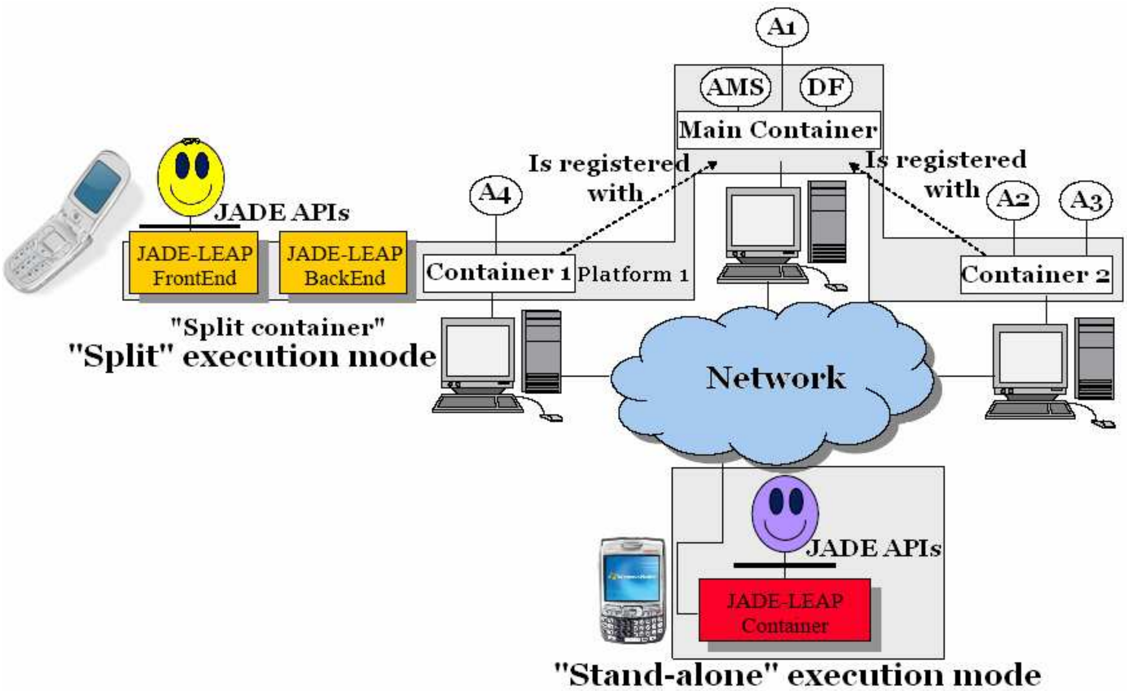

Figure 7: Distributed Ubiquitous Environment Integration

The JADE-LEAP platform helps with the integration of ubiquitous environments. We have developed a sample application to illustrate and test the resources of this platform. Our sample application considers two powerful machines, a limited mobile device and a powerful smartphone device. One powerful machine is running the "Main-Container" of the platform. The other powerful machine represents another computer that belongs to the platform and it is running the "Container-1". Several desktops or notebooks can be integrated in this ubiquitous environment. If the user mobile device is limited by its internal memory capacity, then it has to use the *split* execution mode. This mode allows the device to share resources with another computer that is more powerful.

When the limited device connects with the powerful machine that is running "Container-1", through a wireless network, it requests that a "heavy" part of the container, called "Back-End," be maintained in that powerful machine. The other part, called "Front-End," is lighter than the first and runs in the limited mobile device.

The smartphone is able to run the container using its own resources. Thus, it runs in *stand-alone* execution mode. In this mode the container is located inside the device.

---

6 JADE-LEAP (*JADE Lightweight Extensible Agent Platform*) [Caire 2003].

Using these execution modes, other JADE-LEAP platform resources and the network, we integrate both limited and powerful devices. This integration is one of the central issues in ubiquitous computing. All this process is automatic and the users need not worry about the communication protocols, the execution modes and the platform integration. Our application is self-controlled, based on JADE-LEAP and JADEX infrastructures and agents' collectivity. These technologies avoid the technological gap found on client/server architecture commonly used on ubiquitous solutions to deal with the distributed system concern.

5.3 A Collaborative Ubiquitous Environment centered on the Multi-Agent Systems Paradigm

We argue that the Multi-Agent Systems paradigm is appropriate to model and implement ubiquitous environments, especially when we consider the agents' collaboration, mobility and autonomy properties. To test and verify our hypothesis, we have developed an application based on Multi-Agent System and supported by novel technologies such as JADEX and JADE-LEAP platforms. Figure 8 illustrates this collaborative ubiquitous environment.

The environment is composed of several content servers, users and devices (e.g. mobile phones, PDAs and powerful machines). The application consists of the Multi-Agent System, the JADEX and JADE-LEAP platforms, BDI Model, reasoning and learning techniques and knowledge base. The JADEX-JADE Adapter supports the JADEX and JADE-LEAP platforms communication.

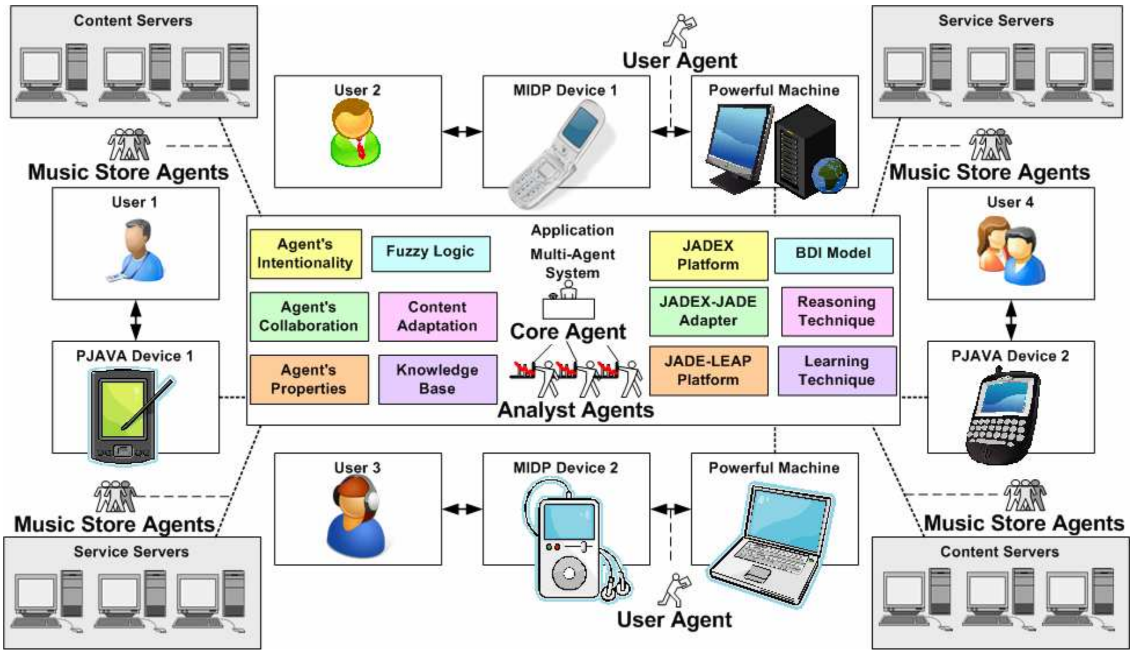

Figure 8: Collaborative Ubiquitous Environment

We centered the application on agent's collaboration and intentionality. Thus, the application agents work together to provide the best service based on the users' requests and preferences. We believe that it is a good way to guarantee the user's satisfaction and to quickly perform the necessary activities.

From the communication perspective, the users can use limited or powerful devices to make the requests. When the device is limited, such as "MIDP Device 1" or "MIDP Device 2," the application is implemented to use the *split* mode execution. To powerful devices such as "PJAVA Device 1" or "PJAVA Device 2" the application uses the *stand-alone* mode execution. Both modes were explained in Section 5.2.

The user can request a specific content (e.g. music or video) through his/her device (e.g. "MIDP Device 1"). We now suppose that the device is limited. Thus, it depends on the powerful machine resources to be integrated to the platform, which is running the application. Part of the platform container will run on the device and part will run on the powerful machine. We implement a simple agent that runs on the devices, even if it is limited. This agent requests the content to *Core Agent*. This last agent, together with other application agents (e.g. Analyst Agents and Music Store Agents), collaborates to provide the best content to the user. The content choice is based on fuzzy logic sets. We use fuzzy logic, reasoning and learning techniques to deal with the different content proposals, which are offered by several content servers. The agent's intentionality is mapped by using a Belief Desire Intention Model (BDI Model). This intention model is detailed in Section 5.4. We create a knowledge base (KB) and some profiles to store and to control the information about user, device and content. Reasoning and learning techniques update the KB and the profiles. Sometimes it is also necessary to adapt the content because of some specific device features.

In this scenario, we basically have four modules to formally describe the developed collaborative ubiquitous environment: (i) *Main Module*; (ii) *JADE-LEAP Module*; (iii) *Profiles Module*; and (iv) *Adaptation Module*. Figure 9 presents these modules.

The *Main Module* is subdivided into the "Reasoning and Learning Module" and the "BDI Module." The first implements the fuzzy logic, reasoning and learning techniques. In this module the main agent is the *Analyst Agent*. The last module contains the BDI Model that considers beliefs, desires and intentions as mental activities to simulate human behavior.

The *JADE-LEAP Module* considers the mobile devices that are limited in terms of internal memory (e.g. MIDP mobile phones).

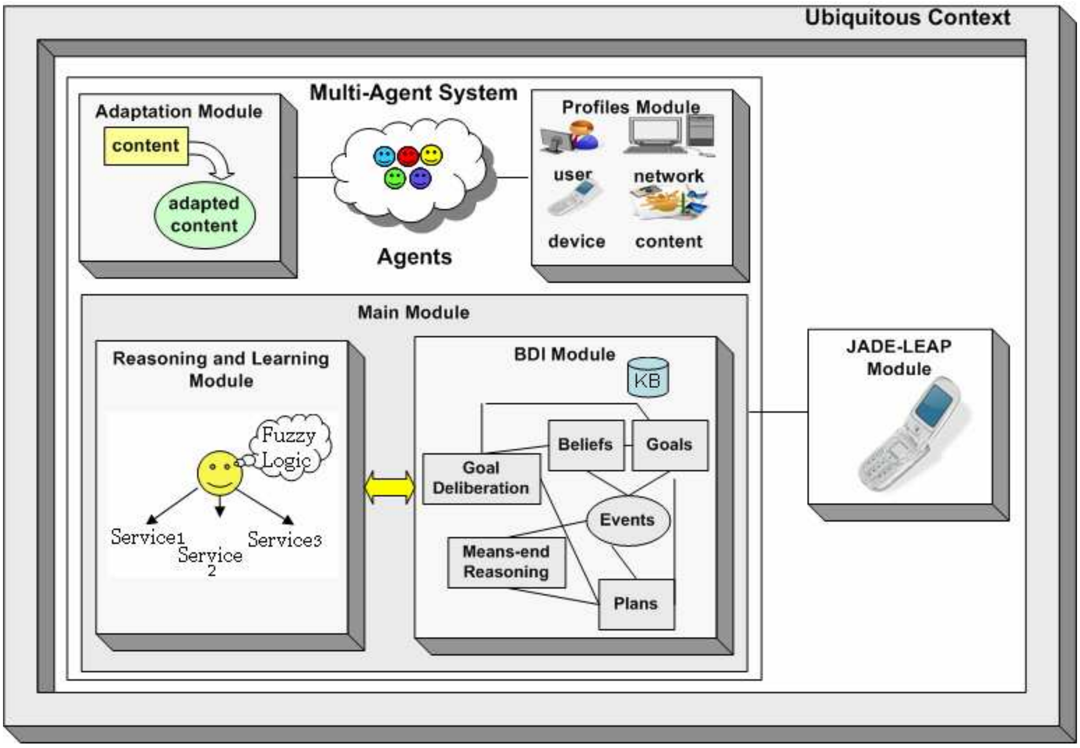

Figure 9: Collaborative Ubiquitous Environment Modules

The *Adaptation Module* is responsible for content adaptation. Sometimes it is necessary to adapt the content to satisfy the user's preferences and to deal with the device capabilities, the content information and the network specifications. In our case study, we dealt with adaptation in the same application server, but we plan to extend it to use a dedicated server to improve the content adaptation. It is a good practice to perform complex content adaptations without overloading the application server.

The *Profiles Module* encompasses the user, device, network and content profiles. Basically, this persistence module provides the information storage, retrieval and updating.

Centered on these modules, our application is generic and can be evolved to represent complex ubiquitous environments, with several heterogeneous devices, service servers and users' preferences. It can also be adequate to deal with other ubiquitous concerns (e.g. content adaptation and context-awareness).

5.4 An Intelligent Ubiquitous Environment through the use of the BDI Model and Fuzzy Logic

We wanted the agents to be able to solve problems such as request receiving and service finding and choice without an expert interaction. For this purpose, we developed a differentiated mechanism based on BDI Model and Fuzzy Logic. This special software infrastructure allowed us to provide an intelligent ubiquitous environment, which is self-controlled according to their agents' goals, beliefs and plans. We illustrate the BDI Model functionalities and their correlations with i* framework elements, through the schema presented on Figure 10.

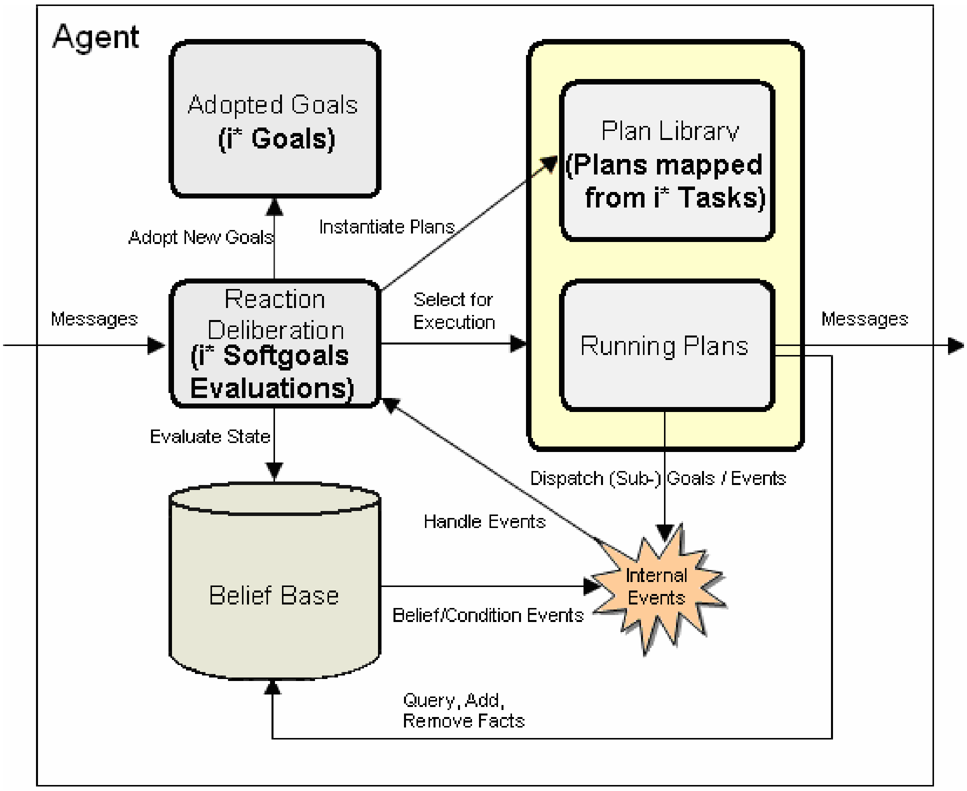

Figure 10: BDI Model

The BDI Model is part of the JADEX platform. This model considers that the agent is based on *Goals*, *Plan Library* and *Belief Base*. The implementation of a reactive and deliberative agent architecture is used to represent mental states following the BDI Model. *Messages* deliberate reactions based on *i* Softgoals Evaluations* that instantiate plans based on *Plan Library*, which is formed by *Plans mapped from i* Tasks*. The plan is selected for execution. New goals (*i* Goals*) can be adopted and the *Belief Base* is consulted to perform a given action. When the plan is running, it dispatches *Goals*, *Subgoals* and *Events*, as *Internal Events* or *Message Events*.

Except for the agents that run on limited devices, based on behavior, all application agents are implemented considering the BDI representation model. Thus, each *Core Agent* has its own knowledge base to consult and to find the content. If content cannot be found by using this strategy, the *Core Agent* delegates the findings and the choices to the *Analyst Agent*. The latter is responsible for analyzing all the proposals offered by other collaborative agents, which consult different knowledge bases. To choose the best service proposal, the *Analyst* uses several fuzzy logic conditional rules and reasoning techniques. In the case of our case study, the *Analyst Agent* analyzes the price, the security and the number of services needed to access the content that are informed by other application agents. Thus, using conditional rules, the *Analyst Agent* establishes the proposal quality values and decides which proposal is the best to satisfy the user's preferences. Finally, the content is informed to *Core Agent*, which is responsible for providing it to the corresponding user.

In our case study, the agents are in charge of all the obligations and application control, as of the moment of the user request to the moment of service offering. This is possible because of the intelligent infrastructure provided by the BDI Model, fuzzy logic, other JADEX resources and the JADE-LEAP execution modes.

The method presented in four phases in Sections 5.1 to 5.4 constitutes an original Software Engineering solution for modeling and developing ubiquitous computing applications based on cognitive Multi-Agent Systems and the concept of Agent's Intentionality.

6 Conclusions and Future Work

In this paper, we argue that Multi-Agent Systems (MAS) can support ubiquitous and personalized interactions based on active and intelligent environments.

In particular, we have presented a development method that produces applications in which a user may delegate her own agent to interact with an environment modeled as a set of agents providing different services and using reasoning techniques. In order to evaluate our ideas, we have developed a platform, which simulates a ubiquitous environment. It makes it possible for the user's personal agent to delegate the service choice to a *Core Agent* according to the desired service and to the current context situation (e.g. user's preferences and device features). This agent decides which service is the best to be offered based on a collaborative Multi-Agent System and several fuzzy conditional rules. We believe that the method and the chosen application illustrated the viability of a practical infrastructure for experimental research in the areas of distributed and mobile environments. They are also generic and easily extendable to represent complex ubiquitous applications with heterogeneous devices, several users and distributed servers.

The results presented in this paper propose solutions to many of the main research and technological challenges of the area of ubiquitous computing, such as distributed environment, different devices and user's satisfaction. We have dealt here with the following challenges: (i) Ubiquitous Requirements Modeling centered on the Agent's Intentionality; (ii) Distributed Ubiquitous Environment Integration using the JADE-LEAP Platform; (iii) Collaborative Ubiquitous Environment centered on the Multi-Agent Systems Paradigm; and (iv) Intelligent Ubiquitous Environment through the use of the BDI Model and Fuzzy Logic.

Finally, we conclude that it is possible to maintain the traceability of software development decisions; to reduce the documentation and the complexity of the commonly used diagrams; and to reuse the decisions (e.g. business rules), the design (e.g. interface templates), the code (e.g. components and patterns), and other artifacts of the ubiquitous application development process. To deal with the distributed environment and several devices, users and service servers, we considered an infrastructure based on technologies such as JADE-LEAP and JADEX platforms, BDI model, Multi-Agent Systems paradigm, goal-oriented planning, fuzzy logic, reasoning and learning techniques.

Although our results are very promising, there are several aspects that we still need to examine during the course of our future work. We present some of them as follows:

- To experimentally compare the results obtained with other methods and platforms;
- To investigate hard topics such as: (i) context-awareness; (ii) dependability; (iii) content adaptation; (iv) governance; (v) reputation, and others.

References

Abowd, G. D.; Mynatt, E. D. “Charting Past, Present, and Future Research in Ubiquitous Computing.” ACM Transactions on Computer-Human Interaction, Vol. 7, Number 1, pp. 29-58, March 2000.

Agotnes, T.; Hoek, W. van der; Wooldridge, M. “Reasoning about Judgment and Preference Aggregation.” Proceedings of Sixth International Conference on Autonomous and Multiagent Systems, AAMAS’07, IFAMAAS, pp.554-561, Honolulu, Hawaii, USA, May 2007.

Alencar, F. M. R., Pedroza, F. P., Castro, J. F. B., and Amorim, R. C. O. “New Mechanisms for the Integration of Organizational Requirements and Object Oriented Modeling.” In: VI Workshop on Requirements Engineering, Piracicaba, pp.109-123, 2003.

Almeida, E.S. “Uma Abordagem Para o Desenvolvimento de Software Baseado em Componentes Distribuídos.” Dissertação de Mestrado, Universidade Federal de São Carlos, São Carlos, SP, 2003.

Andersen, D.; Bansal, D.; Curtis, D.; Seshan, S.; Balakrishnan, H. “System support for bandwidth management and content adaptation in Internet applications.” In Proceedings of 4th Symposium on Operating Systems Design and Implementation, USENIX Association, pp. 213-226, San Diego, CA, October 2000.

Baldauf, M.; Dustdar, S.; Rosenberg, F. “A survey on context-aware systems.” International Journal of Ad Hoc and Ubiquitous Computing, Vol. 2, Number 4, pp. 263 – 277, 2007.

Ballagas, R.; Ringel, M.; Stone, M.; Borchers, J. “iStuff: A Physical User Interface Toolkit for Ubiquitous Computing Environments.” CHI’03, Vol.5, Issue 1, pp. 537-544, FT. Lauderdale, Florida, USA, April 2003.

Bardram, J. E. “The Java Context Awareness Framework (JCAF)-A Service Infrastructure and Programming Framework for Context-Aware Applications.” Pervasive Computing: Third International Conference, PERVASIVE 2005, Munich, Germany, May 2005.

Bauer, B.; Odell, J. “UML 2.0 and agents: how to build agent-based systems with the new UML standard.” Engineering Applications of Artificial Intelligence, Vol. 18, Issue 2, Agent-oriented Software Development, 2005.

Becta, D. L. “Ubiquitous Computing: Emerging Technologies for Learning”. Vol. 2, Chapter 6, pp. 64-79, 2007.

Bell, G.; Dourish, P. “Yesterday’s tomorrows: notes on ubiquitous computing’s dominant vision.” Personal and Ubiquitous Computing, Vol. 11, Number 2, pp. 133-143, February 2007.

Bellifemine, F.; Caire, G.; Greenwood, D. “Developing Multi-Agent Systems with JADE.” John Wiley & Sons, ISBN 0470057475, 286 pages, April 2007.

Bigus, J.; Bigus, J. “Constructing Intelligent Agents Using Java.” Second Edition. Ed. John Wiley & Sons Incorporated, 432 pages, ISBN 978-0471396017, March 2001.

Boudreau, T.; Glick, J.; Greene, S.; Woehr, J.; Spurlin, V. “NetBeans: The Definitive Guide.” O'Reilly Media, Inc., First Edition, ISBN-13: 978-0596002800, 672 pages, October 2002.

Braubach, L.; Lamersdorf, W.; Pokahr, A. “Jadex: Implementing a BDI Infrastructure for JADE Agents.” Distributed Systems and Information Systems, University of Hamburg, Vol.3, n.3, pp.76-85, September 2003.

Bresciani, P., Giorgini, P., Giunchiglia, F., Mylopoulos, J.,Perini, A. “Tropos: An Agent -Oriented Software Development Methodology.” In Autonomous Agents and Multi-Agent Systems, Vol. 8 (3), pp. 203-236, May 2004.

Brumitt, B.; Meyers, B.; Krumm, J.; Kern, A.; Shafer, S. “EasyLiving: Technologies for intelligent environments.” In Proceedings of Second International Symposium on Handheld and Ubiquitous Computing, HUC 2000, Springer Verlag, pp. 12-29, Bristol, UK, September 2000.

Buckley, J.; Mens, T.; Zenger, M.; Rashid, A.; Kniesel, G. “Towards a Taxonomy of Software Change.” Journal on Software Maintenance and Software Evolution: Research and Practice, 2005.

Caire, G. “LEAP User Guide.” Copyright (C) TILAB, LEAP3.1 - Last update: December 2003.

Christopoulou, E.; Kameas, A. “GAS Ontology: An ontology for collaboration among ubiquitous computing devices.” International Journal of Human-Computer Studies, Vol. 62, Issue 5, pp. 664-685, May 2005.

Cunha, H. S. (2007) “Uso de Estratégias Orientadas a Metas para Modelagem de Requisitos de Segurança.” Dissertação de Mestrado. Pontifícia Universidade Católica do Rio de Janeiro, April 2007.

Dubois, E. “A Logic of Action for Supporting Goal-Oriented Elaborations of Requirements.” Proc. 5th International Workshop on Software Specification and

Design, ACM SIGSOFT Software Engineering Notes, Vol. 14, Issue 3, pp. 160-168, Pittsburgh, PA, May 1989.

Garcia, A.; Lucena, C. J. P.; Zambonelli, F.; Omicini, A.; Castro, J. "Software Engineering for Large-Scale Multi-Agent Systems: Research Issues and Practical Applications." Springer, First Edition, ISBN-13: 978-3540087724, Vol. 2602, 285 pages, November 2003.

Gold, N.; Sugden, R. "Theories of team agency." Oxford University Press, Book Chapter, ISBN13: 9780199287260, Number 12, 2008. (To appear in Rationality and Commitment).

Gosling, J.; Joy, B.; Steele, G.; Bracha, G. "Java™ Language Specification." Prentice Hall, 2nd Edition, ISBN-13: 978-0-201-31008-5, 544 pages, June 2000.

Greenfield, A. "Everyware: The Dawning Age of Ubiquitous Computing." Peachpit Press, March 2006.

Harmelen, F. V.; Mcguinness, D. L. "OWL Web Ontology Language Overview." W3C Recommendation, February 2004. Accessed at: http://www.w3.org/TR/2004/REC-owl-features-20040210/ (2008)

Kiczales, G. J.; Lamping, J. O.; Lopes, C. V.; Hugunin, J. J.; Hilsdale, E. A.; Boyapati, C. "Aspect-oriented programming." Patent Number: US 6,467,086 B1, Assignee: Xerox Corporation, Stamford, CT, U.S., July 1999.

Kroll, P.; Kruchten, P. "Rational Unified Process Made Easy – A Practitioner's Guide to RUP." IBM Software Group, Rational Software, Addison-Wesley, 2003.

Lagos, M. E.; Alarcón, R.; Nussbaum, M.; Capponi, F. "Interaction-Based Design for Mobile Collaborative-Learning Software." Pontifícia Universidad Católica de Chile, Software Development Feature, IEEE SOFTWARE, Published by the IEEE Computer Society, pp.80-89, July/August 2007.

Lending, D.; Chervany, N. L. "The Use of Case Tools." Special Interest Group on Computer Personnel Research Annual Conference, Proceedings of the 1998 ACM SIGCPR conference on Computer personnel research, ISBN:0-89791-959-9, pp. 49-58, Boston, MA, USA, 1998.

Liogkas, N.; MacIntyre, B.; Mynatt, E. D.; Smaragdakis, Y.; Tilevich, E.; Voida, S. "Automatic Partitioning: A Promising Approach to Prototyping Ubiquitous Computing Applications." IEEE Pervasive Computing, Special Issue on Building and Evaluating Ubiquitous System Software, March 2004.

Maamar, Z.; Mostefaoui, S. K.; Yahyaoui, H. "Toward an Agent-Based and Context-Oriented Approach for Web Services Composition." IEEE Transactions On Knowledge and Data Engineering, Vol. 17, Number 5, pp. 686-697, May 2005.

Machado R. P.; Ferrão G., Rocha C.; Endler M.; Lucena C. J.P. "Um Serviço de Combinação de Perfis baseado em Ontologias para Colaboração Espontânea entre Usuários Móveis." ANAIS do VI Workshop de Comunicação Sem Fio e Computação Móvel – WCSF 2004, Gráfica UNIFOR, Fortaleza, pp. 137-142, 2004.

McBurney, P.; Parsons, S. "Retraction and Revocation in Agent Deliberation Dialogs." Springer Netherlands, Vol. 21, Number 3, pp. 269-289, September 2007.

Mylopoulos, J.; Chung, L.; Nixon, B. "Representing and Using Non-Functional Requirements: A Process-Oriented Approach." IEEE Trans. Soft. Eng., 18 (6), June 1992.

O'Hare, G. M. P.; O'Grady, M. J. “Gulliver's Genie: a multi-agent system for ubiquitous and intelligent content delivery .” Computer Communications, Vol. 26, Issue 11, pp.1177-1187, July 2003.

Pokahr, A. and Braubach. L. “JADEX Tool Guide.” Distributed Systems Group, University of Hamburg, Germany, Release 0.96 - Last Update: June 2007 [a]. Accessed at: http://vsis-www.informatik.uni-hamburg.de/projects/jadex (2008)

Pokahr, A. and Braubach. L. “JADEX Tutorial.” Distributed Systems Group, University of Hamburg, Germany, Release 0.96 - Last Update: June 2007 [b]. Accessed at: http://vsis-www.informatik.uni-hamburg.de/projects/jadex (2008)

Pokahr, A. and Braubach. L. “JADEX User Guide.” Distributed Systems Group, University of Hamburg, Germany, Release 0.96 - Last Update: June 2007 [c]. Accessed at: http://vsis-www.informatik.uni-hamburg.de/projects/jadex (2008)

Pokahr, A.; Braubach, L.; Lamersdorf, W. “Jadex: A BDI Reasoning Engine.” Programming Multi-Agent Systems, Kluwer Academic Publishers, pp.149-174, 2005.

Pressman, R.S. “Software Engineering: A Practitioner’s Approach.” Mcgraw-Hill Series in Computer Science, Sixth Edition, ISBN-13: 978-0072853186, 896 pages, March 2004.

Satoh, I†. “Mobile Applications in Ubiquitous Computing Environments.” Copyright The Institute of Electronics, Information and Communication Engineers, SATOH: Mobile Applications in Ubiquitous Computing Environments, IEICE TRANS. COMMUN., Vol.E88–B, n.3, pp.1026-1033, March 2005.

Schulzrinne, H.; Wu, X.; Sidiroglou, S. “Ubiquitous Computing in Home Networks.” IEEE Communications Magazine, pp. 128-135, November 2003.

Silva, V. T.; Lucena, C. J. P. “From a Conceptual Framework for Agents and Objects to a Multi-Agent System Modeling Language.” Journal of Autonomous Agents and Multi-Agent Systems, Kluwer Academic Publishers, 2003.

Silva, V. T.; Lucena, C. J. P. “From a Conceptual Framework for Agents and Objects to a Multi-Agent System Modeling Language.” Thesis, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), Rio de Janeiro, 2004.

Soldatos, J.; Pandis, I.; Stamatis, K.; Polymenakos, L.; Crowley, J. L. “Agent based middleware infrastructure for autonomous context-aware ubiquitous computing services.” Special Issue: Emerging Middleware for Next Generation Networks, Vol. 30, Issue 3, pp. 577-591, February 2007.

Wallace, M.; Novello, S.; Schimpf, J. “ECLiPSe: A Platform for Constraint Logic Programming.” Technical report, IC-Parc, Imperial College, 32 pages, London, 1997. Accessed at: http://citeseer.ist.psu.edu/wallace97eclipse.html (2008)

Yau, S. S.; Karim, F. “Context-Sensitive Middleware for Real-Time Software in Ubiquitous Computing Environments.” Fourth International Symposium on Object-Oriented Real-Time Distributed Computing, p. 0163, 2001.

Yu, E. S. K. “Modelling Strategic Relationships for Process Reengineering”. PhD. Thesis, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada, 1995.

Yu, E. S. K. “Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering.” Faculty of Information Studies, University of Toronto, Toronto, Ontario, Canada, 1998.

Zambonelli, F., Jennings, N. R. and Wooldridge, M. "Developing Multiagent Systems: the Gaia Methodology." ACM Trans on Software Engineering and Methodology, Vol.12(3), pp. 317-370, 2003.