PUC

ISSN 0103-9741

Monografias em Ciência da Computação
n° 52/08

The Modeling and Implementation of Ubiquitous Applications based on Agents' Intentionality

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. 52/08 ISSN: 0103-9741
Editor: Prof. Carlos José Pereira de Lucena December, 2008

The Modeling and Implementation of Ubiquitous Applications based on Agents' Intentionality

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

[1] Departamento de Informática – PUC-Rio
Rua Marques de São Vicente 225, 13º andar Ed. Padre Leonel Franca.
Gávea, 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 computing application area has issued original challenges, such as different devices 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 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 consider 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 exemplar applications centered on agent's intentionality and the Belief Desire Intention Model, that results in a specific Software Engineering method for the modeling and implementation of ubiquitous applications.

Keywords: Ubiquitous Computing, Distributed Intentionality, BDI Model, Multi-Agent Systems.

Resumo. As abordagens atuais para computação distribuída precisam ser aprimoradas e estendidas para o domínio da Computação Ubíqua. Essa nova área da computação tem desafios próprios, tais como diferentes dispositivos e serviços. Logo, uma abordagem apropriada é necessária para coordenar computadores heterogêneos, que atuam como um computador virtual cercado de ambientes ubíquos distribuídos, marcados pela mobilidade. Além disso, essa abordagem tem que suportar vários serviços que estão além da capacidade de computadores simples. Nesse cenário, as possibilidades oferecidas pelo uso de Sistemas Multi-Agentes (SMA) bem engenheirados parecem ser pertinentes. Visando considerar formas apropriadas para lidar com os desafios impostos na Computação Ubíqua e investigar métodos e técnicas de software emergentes para esse domínio, estamos desenvolvendo aplicações experimentais centradas em intencionalidade de agentes e no modelo BDI (*Belief Desire and Intention*), que resultaram em um método de Engenharia de Software específico para modelagem e implementação de aplicações ubíquas.

Palavras-chave: Computação Ubíqua, Intentionalidade Distribuída, Modelo BDI, Sistemas Multi-Agentes.

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 [1] [2] [3]. 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 authors Gregory D. Abowd and Elizabeth D. Mynatt, who argue in [4]:"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 [4] [5] [6], 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 (e.g. smartphones, palms, notebooks, and desktops) 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 monograph, we believe that Multi-Agent Systems is an appropriate approach for addressing applications that work in a distributed setting and handle different devices running different 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 [7] [8] [9] can be used to search, to choose and to offer the best service to a user's request; and (ii) the agent can be viewed 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 applications centered on collaborative Multi-Agent Systems. We use several MAS concepts such as agent's intentionality [10] [11] and the Belief Desire Intention Model (BDI Model) [12]. We believe that our goal orientation approach is an improvement over other proposal for modeling ubiquitous applications.

Furthermore, we describe a novel Software Engineering method for Ubiquitous Computing modeling and implementation centered on Multi-Agent Systems and agent's intentionality. Subsequently, we also organized and documented 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 monograph 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 Studies. 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 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 (JADE Lightweight Extensible Agent Platform) [13] 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:** One of the main goals of ubiquitous computing is to provide the service to the user with quality, security, and anywhere and anytime. Thus, it is necessary to consider different user's interests. Our proposal is to use a collaborative MAS to deal with this situation. Some reasoning and learning techniques combined with ubiquitous profiles 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 profiles combined with JADE-LEAP execution modes (see Section 5) to deal with this problem.
- **Several Services and Contents:** Our research focuses on users that request different services and content. Thus, we had to consider several remote servers in our experimental ubiquitous environment.

Only to contextualize, a typical ubiquitous environment is composed of a group of people with specific interests. 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 users' satisfaction, heterogeneous devices and several services. Moreover, the services omnipresence is a fact in this context.

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 1 shows the three categories of methods, tools and properties.

The Traditional Applications Need [14] involves specific paradigms (e.g. Object Orientation, Aspect, Components, and Multi-Agent Systems [15]), methodologies [16] [17], technologies (Corba, Web-Service, Fuzzy Logic), taxonomy [18], and powerful computers (Desktops). Our work investigates these solutions to identify if they are appropriate to be used for ubiquitous computing. Moreover, we analyze some Ubiquitous Applications General Needs such as reasoning and learning techniques, heterogeneous devices (e.g. cellular phone, smartphone and PDAs) and smart-spaces [19].

Finally, we also investigate Ubiquitous Applications Specific Needs such as mobility [20], adaptability [21], context awareness [22], and profiles [23] (e.g. user, contract, network, content and device profiles) to guarantee users' satisfaction.

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 for 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 and implementation approach centered on Multi-Agent Systems and agent's intentionality.

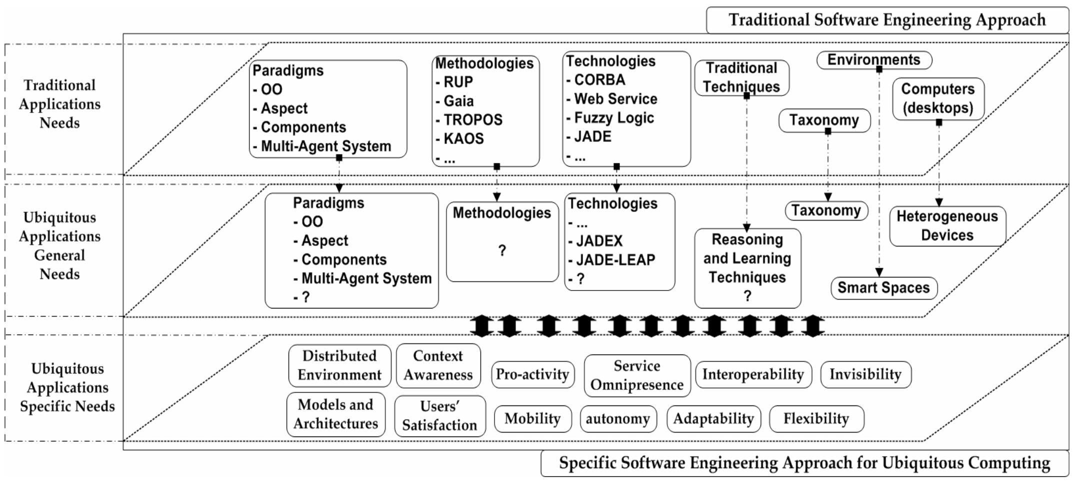

Figure 1: Traditional Software Engineering and Software Engineering for Ubiquitous Computing

3 Some Related Work

Several research efforts have proposed solutions to deal with Software Engineering problems, such as formalisms and languages. However, only a few of them address specific problems raised by ubiquitous applications development (from requirements to code). 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 monograph.

According to [24], one of the main concerns facing ubiquitous computing development is the difficulty of writing software for complex, heterogeneous and distributed applications. Liogkas et al. 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 adequate experimental environment using agents' intentionality and other State-Of-The-Art technologies (e.g. JADE-LEAP). Our environment may help the software engineer to deal with complex applications. We also argue that our methods and tools are widely applicable as presented in our case studies (see Section 4).

The work presented in [2] [25] [26] particularly motivated us to develop an environment based on the collaboration and interaction properties. They 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. In [2], iStuff is a physical user interface toolkit for ubiquitous environment. This toolkit and a flexible software framework to support it were proposed by Ballagas et al. to simplify the interaction techniques between multiple users, devices and applications. The developed environment is collaborative and interactive. This software infrastructure includes a dynamically configurable support for the mapping of devices to applications. In [25], 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. The research is based on beliefs, desires and intentions, which are used to determine the base deductions. Finally, the paper [26] 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. Maamar et al. also discuss the security of the computing resources. However, although [2] [25] [26] are relevant research efforts for ubiquitous computing, the infrastructure provided by these proponents does not have support for the task of building a software within an engineering approach. Our contribution fills this gap, offering an adequate support for this purpose.

Other work that is been used in our current work is presented in [27] [3] [28] [29] [30]. In [27], Schulzrinne et al. 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 [3], Reconfigurable Context-Sensitive Middleware (RCSM) is presented to facilitate real-time context-aware software in ubiquitous computing environments. The researchers Yau and Karim argue that context awareness is increasingly becoming an important capability in devices for ubiquitous computing environments. The paper [28] is centered on middleware for ubiquitous and context-aware computing. Soldatos et al. 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 [29] 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. In [30], Becta 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, for example, providing support for the applications security issue. Moreover, in most cases, this research contributes to the implementation of ubiquitous application and is not usually concerned about the requirements and design activities. In order to fill this technological gap, 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 monograph is in part motivated by the fact that there are few known specific approaches for ubiquitous computing from the Software Engineering perspective. 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 Studies Overview

We developed Multi-Agent applications based on the agent's intentionality [10] [11] and collaboration to investigate the proposed research issues and a specific Software Engineering method to support the development of Ubiquitous Computing from the requirements to implementation.

4.1 First Case Study (Media Shop Project)

Our first case study simulates a ubiquitous experimental environment with different devices, users and services. We consider heterogeneous 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, represented by notebooks and desktops with high processing capacities, assist the communication between limited mobile devices and the application to be developed. The contents and services, which are requested 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 platform [13], JADEX framework [31]), and reasoning [7] [8] [9] and learning techniques [7] [32] to provide the best service based on the user's preferences. They include the device features, service/content price, server security, contract and other parameters.

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 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 [12] to control the communication of the agents, which use collectivity, collaboration, autonomy, mobility and other properties to perform their activities driven by their intentionality.

1 Mobile Information Device Profile such as simple mobile phones.

2 Personal JAVA devices such as smartphones.

4.2 Second Case Study (DIUP Project)

We developed another ubiquitous case study called DIUP (Dental Informatics Ubiquitous Project). This project is based on the automation of some dentist's task. The main purpose is to improve dental care to deprived communities through the use of heterogeneous mobile devices (e.g. smartphones and palms). In this context, the mobile devices can allow the performing of important activities (e.g. patient registration, dental diagnosis, drug prescriptions and oral diseases searching) intrinsic to dental treatment anywhere and anytime. A collaborative and intelligent Multi-Agent System is used to manage and to perform these activities, and to achieve users' goals. We extended our framework for content adaptation, called IFCAUC [33] (Intentional Framework for Content Adaptation in Ubiquitous Computing), to improve the DIUP development considering the adaptation transversal issue. In this scenario, we adapt different kind of contents (e.g. x-ray images) using, for example, transcoding and resizing. Moreover, we explored some important agents' properties (e.g. mobility, flexibility, adaptability and autonomy) to perform the adaptations on dedicate servers according to ubiquitous profiles that were stored in a knowledge base.

The experiments, which were implemented to develop and to validate our case studies, 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 MAS

Our first research results, based on our experimental laboratory (Media Shop and DIUP), 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; (iv) to build an intelligent ubiquitous environment; and (v) to test the ubiquitous environment using an experimental validation with the users' collaboration. The ideas are described in Sections 5.1 to 5.5.

5.1 Modeling centered on the Agents' Intentionality

In order to model the ubiquitous requirements, we propose the use of the i* framework [34] that models the agents based on their intentionality. The advantages of using this framework, which were observed during our research, are: (1) the reduction of the number of commonly used diagrams [35] [36], since the proposed i* modeling suggests the use of only two models (described as follows) to represent the functional and non-functional requirements; and (2) the modeling of functional (e.g. goals, tasks, subgoals and subtasks) and non-functional (e.g. softgoals) requirements centered in the Goal-Orientation [37] that is believed to increase the amount of explicit traces within the MAS model. It is important to consider that traces are fundamental for dealing with the ever evolving requirements.

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.

A very simple example of an SD model based on our first case study, Media Shop Project explained on Section 4.1, is presented in Figure 2. The User Agent depends on the Core Agent to choose the service. It does not matter how the Core Agent will achieve this state. Thus, it is a Goal Dependency. The Core Agent depends on the User Agent 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 Chosen"), the resources ("User's Information" and "Service Information") and the dependencies relationships between the User Agent and the Core Agent.

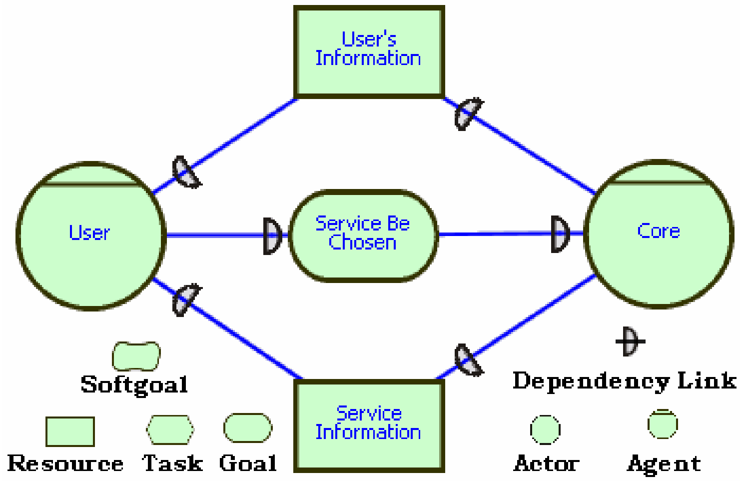

Figure 2: Agent-Agent SD Model

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, since 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 3 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.

3 The depending agent is called depender [38].
4 The agent who is depended upon is called dependee [38].

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).

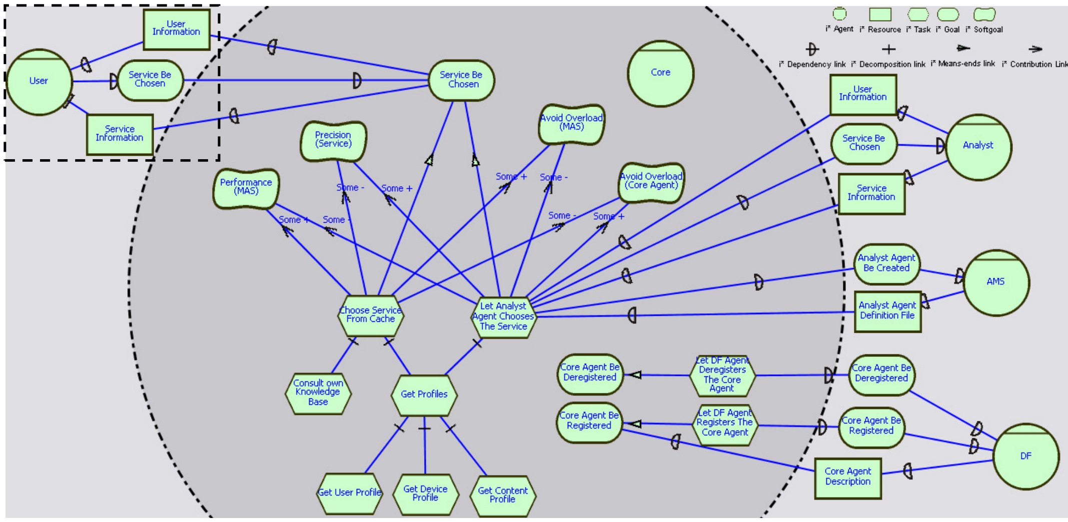

Figure 3: Core Agent Strategic Rationale Model

5.2 A Distributed Ubiquitous Environment Integration

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 4 shows the execution modes.

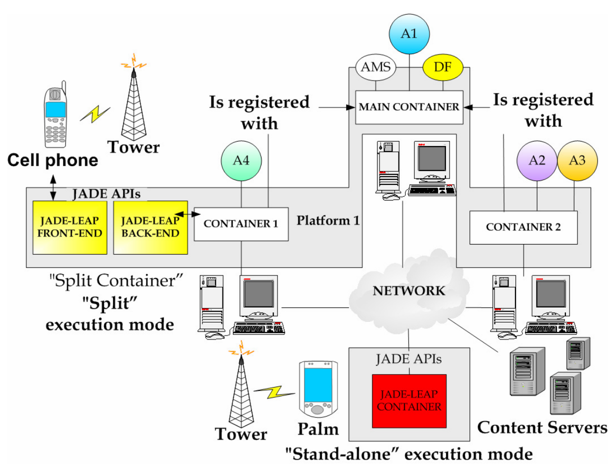

Figure 4: Distributed Environment Integration

5 In our context, the term "satisficed" is quite different from "satisfied". It means "satisfied to a sufficient degree". This term was borrowed from H. Simon and has been used by the NFR literature [39].
6 JADE Lightweight Extensible Agent Platform [13].

The JADE-LEAP platform helps with the integration of ubiquitous environments. We have developed a typical ubiquitous scenario, implemented in our two case studies, to illustrate and test the resources of this platform. Our ubiquitous scenario considers two powerful machines, and limited (Cell Phone) and powerful (Smartphone) mobile devices. 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's 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.

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. The entire process is automatic and the users need not worry about the communication protocols, the execution modes and the platform integration. We can say that this process is invisible to the user. The process complexity invisibility is one of the desired features in ubiquitous applications according to Mark Weiser in [40]. Our applications are 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 MAS

We argue that the Multi-Agent Systems paradigm is appropriated 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 applications based on Multi-Agent Systems and supported by novel technologies such as JADEX framework and JADE-LEAP platform.

The environments are composed of several content servers, users and devices. The applications consist of the MAS, JADEX, JADE-LEAP, BDI Model, reasoning and learning techniques, knowledge bases and others frameworks used to deal with transversal issues such as adaptability. The JADEX-JADE Adapter supports the JADEX and JADE-LEAP communication.

We centered the applications on agents' collaboration and intentionality. Each application's agents work together to provide the best service based on the users' 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 devices, the applications are implemented to use the split execution mode. For powerful devices, such as Personal JAVA devices, the applications use the stand-alone execution mode. Both modes were explained in Section 5.2.

In the Media Shop case study, the user can request a specific content (e.g. music or video) through his/her device. We now suppose that the device is limited. Thus, it depends on the powerful machine resources to be integrated into the platform that 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 latter 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 the 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 shown in the DIUP case study.

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

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 have limited internal memory. This module is represented by a simple agent, which is implemented based on its behaviors. 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 Media Shop Project, we dealt with adaptation in the same application server, and in DIUP Project, we used a dedicated server to adapt the content. The latter solution is a good practice for performing 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 applications are generic and can be evolved to represent complex ubiquitous environments, with several heterogeneous devices, service servers and users' preferences. It may also be adequate to deal with other ubiquitous concerns, which will be investigated in further work.

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

We wanted the software 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 its agents' goals, beliefs and plans. We illustrate the BDI Model functionalities and their correlations with i* framework elements, through the schema presented in Figure 5.

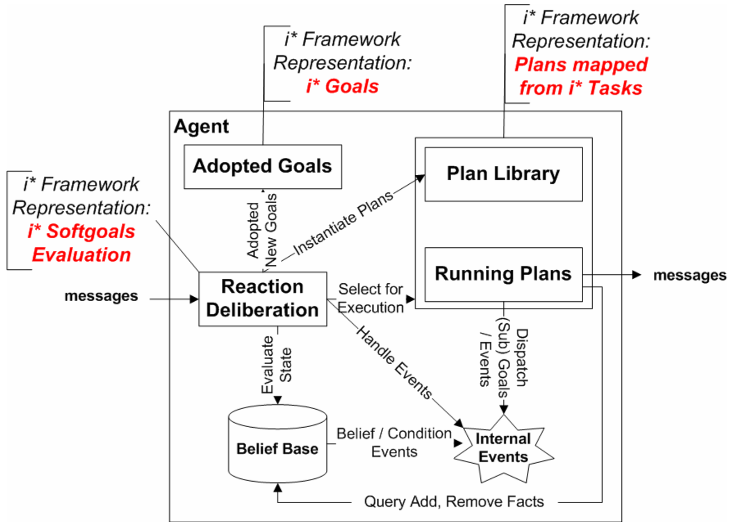

Figure 5: BDI Model

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 applications' agents are implemented considering the BDI model. In the Media Shop Project, 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 based on its beliefs, desires and intentions. The variables used in the conditional rules are organized in: (i) Fuzzy Input Variables such as Price with 6 fuzzy sets; and (ii) a Fuzzy Output Variable as Quality with 7 fuzzy sets. Figure 6 shows the fuzzy sets for the Price variable.

```
// create price continuous variable
ContinuousFuzzyRuleVariable price =
   new ContinuousFuzzyRuleVariable(sfrb, "price", 0.00, 10.00);
price.addSetShoulder("free", 0.1, 0.25, 0.75, FuzzyDefs.LEFT);
price.addSetTrapezoid("cheap", 0.1, 0.50, 1.00, 1.25, 2.00);
price.addSetTrapezoid("medium", 0.1, 1.00, 1.75, 2.25, 3.00);
price.addSetTriangle("high", 0.1, 2.00, 2.75, 3.50);
price.addSetTrapezoid("expensive", 0.1, 2.50, 3.50, 4.00, 4.50);
price.addSetShoulder("abusive", 0.1, 4.00, 4.50, FuzzyDefs.RIGHT);
```
Figure 6: Fuzzy Rule Variable Price

In the Media Shop Project, presented on Section 4.1, the Analyst Agent analyzes the price, the security and the number of services needed to access the content that is 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.

However, as already explained before, sometimes it is necessary to adapt the content. In the DIUP Project, presented on Section 4.2, the image of a x-ray had frequently to be adapted to the device specifications (ex. accepted media type; memory capacity and screen resolution). Thus, the Core Agent delegates this adaptation to Adaptation Mediator Agent that requests the creation of a Mobile Adapter Agent to move to another smart-space (adaptation server) to adapt the content. The Adaptation Mediator Agent also uses fuzzy logic and its intentionality to choose an adequate adaptation server. The Mobile Adapter Agent migrates to adaptation server and collaborates with the Adapter Agent to perform the necessary adaptation (e.g. transcoding or resizing). Thus, the Mobile Adapter Agent returns to the application server with the adapted content, updates the knowledge base and informs the Adaptation Mediator Agent about the content. The Adaptation Mediator Agent communicates with the Core Agent. The content is provided to the corresponding user.

In our case studies, 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.

5.5 Using Experimental Validation to Test the Ubiquitous Environment

We performed some tests that demonstrated to us that the proposed solutions presented in this monograph are very promising to deal with important Ubiquitous Computing issues. Firstly, some volunteers (ten recently graduated dentists) were invited to evaluate the user's satisfaction using the DIUP application.

Each dentist specified her/his preferences (e.g. desired image type) and personal data (e.g. contract information) that were stored in the user profile. Additionally, the dentists had to use different devices (e.g. cellular and palm). Information about each device is also registered in the device profile. Other important data is stored in network and content profiles.

Finally, each professional requested a content using her/his mobile device. The DIUP application returned the desired images based on the use of different adaptation strategies, which are centered on user's satisfaction and ubiquitous profiles. Thus, the volunteers evaluated her/his satisfaction according to the obtained image. Figure 7 illustrates the results.

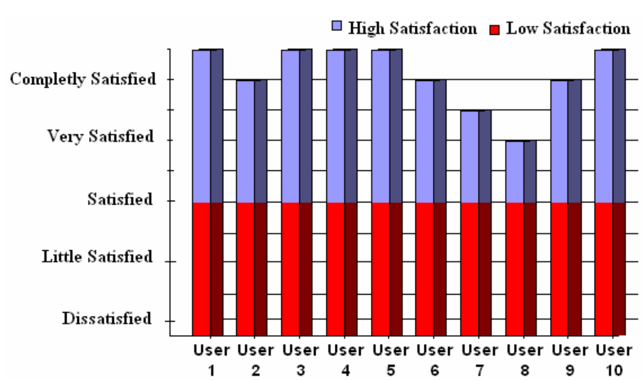

Figure 7: Users' Satisfaction (DIUP Project)

We obtained high users' satisfaction in relation to the quality and the pertinence of the image. We also observed that the volunteers, who were not fully satisfied, obtained images compatible with their access devices. In these rare cases, according to the limitations of these devices, it was not possible to adapt the image based on the user's preferences. However, these dentists evaluated themselves as "Very Satisfied." They based their evaluation on the high quality of the image and on the low waiting time (immediately) to receive it.

Furthermore, we analyzed the agents' performances, considering different adaptations (e.g. transcoding and resizing), which were necessary according to the images requested by the volunteers. The results are very promising. It is important to emphasize that the requested images had different and specific resolutions and adaptation needs. Figure 8 shows an example of a log, which was obtained with the tests.

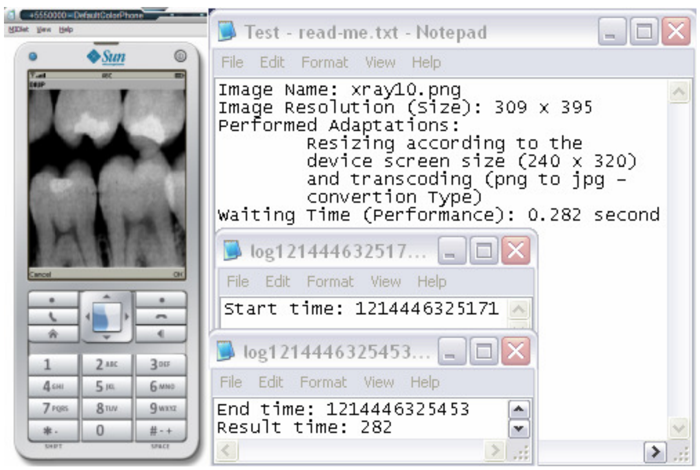

Figure 8: Agents' Performance (DIUP Project)

The method presented in five phases in Sections 5.1 to 5.5 constitutes an original Software Engineering solution for modeling, developing and testing ubiquitous computing applications based on cognitive Multi-Agent Systems and the concept of agent's intentionality. The agent infrastructure presented in this monograph can also be viewed as the nucleus of a product line for the ubiquitous software development centered on MAS. This product line is improved in our approach by using middleware and frameworks (e.g. IFCAUC and Hibernate) to deal with transversal issues (e.g. adaptation and persistence needs).

6 Conclusions and Future Work

In this monograph, we argue that 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 an own agent to interact with an environment modeled as a set of agents providing different services. In order to evaluate our ideas, we have developed applications, which simulate ubiquitous environments. It makes it possible for the user's personal agent to delegate the service choice or the content adaptation to a cognitive 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 or which server is more adequate to perform the content adaptation based on a collaborative MAS and several fuzzy conditional rules.

We believe that the method and the chosen case studies illustrated the viability of an infrastructure for experimental research in the areas of distributed and mobile environments.

The results presented in this monograph brings alternatives to following ubiquitous technological challenges: (i) ubiquitous requirements modeling centered on the agent's intentionality; (ii) distributed ubiquitous environment integration using the JADELEAP; (iii) collaborative ubiquitous environment centered on MAS; (iv) intelligent ubiquitous environment through the use of the BDI Model and fuzzy logic; and (v) ubiquitous application tests based on experimental validations. Finally, we conclude that the modeling based on distributed intentionality allows maintaining the traceability of software development decisions to reduce the documentation and the number of commonly used diagrams; and to reuse the decisions and the design solutions.

Although our results are very promising, there are several aspects that we still need to examine during the course of our future work. Some of them are: (a) to experimentally compare the results obtained with other technologies; and (b) to investigate hard topics such as context-awareness using ontology; dependability; governance; reputation, and others.

References

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

[2]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, Lauderdale, Florida, USA, April 2003.

[3]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.

[4]Abowd, G. D.; Mynatt, E. D. “Charting Past, Present, and Future Research in Ubiquitous Computing.” ACM Trans. On Computer-Human Interac., Vol. 7, N. 1, pp. 29-58, 2000.

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

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

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

[8]Agotnes, T.; Hoek, W. van der; Wooldridge, M. “Reasoning about Judgment and Preference Aggregation.” Sixth International Conference on Autonomous and Multiagent Systems, pp.554-561, Hawaii, USA, May 2007.

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

[10]McBurney, P.; Parsons, S. “Retraction and Revocation in Agent Deliberation Dialogs.” Springer Netherlands, Vol. 21, N. 3, pp. 269-289, September 2007.

[11]Gold, N.; Sugden, R. “Theories of team agency.” Oxford University Press, Book Chapter, ISBN13: 9780199287260, N. 12, 2008.

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

[13]Caire, G. “LEAP User Guide.” V3.1, December 2003.

[14]Pressman, R.S. “Software Engineering: A Practitioner's Approach.” McGraw-Hill in Computer Science, Edition 6, ISBN-13: 978-0072853186, 896 pages, March 2004.

[15]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.

[16]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.

[17]Bresciani, P., Giorgini, P., Giunchiglia, F., Mylopoulos, J.,Perini, A. “Tropos: An Agent-Oriented Software Development Methodology.” AAMAS, Vol. 8 (3), pp. 203-236, May 2004.

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

[19]Brumitt, B.; Meyers, B.; Krumm, J.; Kern, A.; Shafer, S. “EasyLiving: Technologies for intelligent environments.” II International Symposium on Handheld and Ubiquitous Computing, HUC, pp. 12-29, Bristol, UK, September 2000.

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

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

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

[23]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.” VI Workshop de Comunicação Sem Fio e Computação Móvel (WCSF), pp. 137-142, Fortaleza, 2004.

[24]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, Issue: Building and Evaluating Ubiquitous System Software, March 2004.

[25]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.

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

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

[28]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.

[29]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.

[30]Becta, D. L. “Ubiquitous Computing: Emerging Technologies for Learning”. Vol. 2 (6), pp. 64-79, 2007.

[31]Pokahr, A. and Braubach. L. “JADEX Tutorial.” Distributed Systems Group, University of Hamburg, Germany, Release 0.96, June 2007.

[32]Lagos, M. E.; Alarcón, R.; Nussbaum, M.; Capponi, F. “Interaction-Based Design for Mobile Collaborative-Learning Software.” PUC-Chile, Software Development Feature, IEEE Computer Society, pp.80-89, August 2007.

[33]Serrano, Milene; Serrano, Maurício; Lucena, C. J. P. “Framework for Content Adaptation in Ubiquitous Computing centered on Agents' Intentionality and Collaborative MAS.” IV Workshop on Software Engineering for Agent-oriented Systems, 12 pages, 2008.

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

[35]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.

[36]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.

[37]Dubois, E. “A Logic of Action for Supporting Goal-Oriented Elaborations of Requirements.” 5th International Workshop on Software Specification and Design, ACM SIGSOFT, Vol. 14, Issue 3, pp. 160-168, May 1989.

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

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

[40]Weiser, M. “Some Computer Issues in Ubiquitous Computing.” ACM SIGMOBILE Mobile Computing and Communications, New York, NY, USA, 1999.