UBIQUITOUS SOFTWARE DEVELOPMENT DRIVEN BY AGENTS' INTENTIONALITY

Milene Serrano, Maurício Serrano and Carlos José Pereira de Lucena
Department of Informatics, Pontifical Catholic University of Rio de Janeiro (PUC-Rio)
Rua Marques de São Vicente 225, 13º andar, Ed. Padre Leonel Franca, Gávea, Rio de Janeiro, Brazil

Keywords: Ubiquitous Software Development, Agents' Intentionality, Collaborative Multi-Agent Systems, Distributed Intentionality Modeling, Belief Desire Intention Model.

Abstract: Ubiquitous computing is a novel computational paradigm in which the users' mobility, the devices' heterogeneity and the service omnipresence need is intrinsic and intense. In this context, the ubiquitous software development poses some particular challenges that are not yet dealt with by the traditional approaches found in the Software Engineering community. In order to improve the ubiquitous software development, this paper describes a detailed technological set based on multi-agent systems (MAS), goal-orientation, the BDI (Belief Desire Intention) model and various frameworks and conceptual models.

1 INTRODUCTION

Nowadays, the advances of the Internet, the wireless network, anywhere and anytime information dissemination and device heterogeneity are contributing to the unfolding and growth of studies in Ubiquitous Computing. This novel paradigm represents an expansion challenge for Software Engineering professionals.

Ubiquitous Computing, introduced by Mark Weiser and published in 1991, idealizes a world in which the main requirements are the users' satisfaction, the service omnipresence and the process complexity invisibility to provide different services. In this distributed and untapped context, it is important to investigate traditional and emergent technologies that improve the ubiquitous software development. Some particular challenges of ubiquitous applications are: the inherent users' mobility; the need to consider different devices and the integration of a number of environments.

Ubiquitous software development poses some particular challenges, such as those presented previously, which are not yet dealt with by the traditional approaches found in the Software Engineering community. In order to improve the ubiquitous software development, this paper describes a detailed technological set based on collaborative MAS, goal-orientation, the BDI model and various frameworks and conceptual models. Our main contribution is to purpose a systematic way to develop ubiquitous software centered on intentional frameworks and conceptual models reuse.

Additionally, as we are considering the Ubiquitous Computing and the MAS paradigms, it increases the importance of investigating some special concerns. For example, the agents in ubiquitous applications must be based on context to adapt their actions according to real world situations. The BDI model (Pokahr et al. 2005) can be viewed as an emergent model to develop software agents based on their intentionality. The intentional agents reason, learn and take decisions according to their beliefs, desires and intentions. The beliefs can be stored in the agent's knowledge base, and represent what the intentional agent believes that the real world is. The desires are the goals to be achieved. The intentions are the strategic plans to achieve the goals. The BDI model can be improved by using fuzzy logic, as shown in this paper, which significantly increases agents' cognition. Thus, a new approach for the agent-oriented ubiquitous software engineering has to be considered.

This paper is organized in sections. Section 2 shows the related work. Section 3 describes some ubiquitous challenges. Section 4 presents our proposal. Finally, Section 5 summarizes the conclusions.

Serrano M., Serrano M. and Pereira de Lucena C. (2009).
UBIQUITOUS SOFTWARE DEVELOPMENT DRIVEN BY AGENTS' INTENTIONALITY.
In *Proceedings of the 11th International Conference on Enterprise Information Systems - Software Agents and Internet Computing*, pages 25-34
DOI: 10.5220/0001857400250034

2 RELATED WORK

According to (Saha et al. 2003), Ubiquitous Computing is an interesting domain that is investigated by various research groups: **(i)** *Aura (Carnegie Mellon University)*: the main objective is providing easy access to different services anywhere and anytime; **(ii)** *Endeavour (University of California in Berkeley)*: the main objective is allowing the user interaction with different devices, people, and information; **(iii)** *Oxygen (MIT)*: the main objective is naturally incorporating the communication technologies in the people life; **(iv)** *Potolano (University of Washington)*: the main objective is creating a testing environment for the investigation of Ubiquitous Computing challenges using specific devices for each task.

Additionally, there are several research projects that consider different concerns of Ubiquitous Computing, such as the content adaptation and the context awareness. In (Ravindran et al. 2002), the authors emphasize the importance of the service personalization concept to improve the content adaptation in ubiquitous scenarios. It consists of delivering personalized services based on some specific information, stored in a profile. In order to support this personalization, they propose a framework for managing service personalization in a domain; and the results are documented in a survey that particularly helped us to improve our work. In (Hightower and Borriello 2001), the authors present a set of basic techniques used for location-sensing, documented in a survey. Moreover, they define how the taxonomy, defined by them, can be used to evaluate location-sensing systems.

Despite all the efforts to propose support for Ubiquitous Computing, there is a lack of approaches that consider the ubiquitous software development from the requirements to code. Our main contribution is offering a set of technological supports to deal with some ubiquitous applications challenges and to orient the software engineer in the systematic development of ubiquitous software. We documented our approach as a specific ubiquitous profile (using the building blocks concept) based on the TROPOS (Giunchiglia et al. 2002) disciplines and the RUP phases (Kruchten 2003). Details are presented in Section 4.

3 UBIQUITOUS CHALLENGES

Some challenges commonly found in the ubiquitous software development are: **(i)** *Heterogeneous Network*: the network can be wire or wireless, and it can obey different communication protocols (e.g. DQDB (Distributed Queue Dual Bus), SMDS (Switched Multimegabit Data Services) and SONET (Synchronous Optical Network)). Thus, it is important to establish ways to allow the users' communication in this heterogeneous medium; **(ii)** *Geography and Network Topology*: the security in the communication with heterogeneous mobile devices is quite different when we analyze it using a physical or virtual perspective. In Ubiquitous Computing using the geographical proximity instead of using the network topology is preferred. The former guarantees more security and direct communication; **(iii)** *Limited Battery of the Devices*: as the device is dependent on the battery to perform its basic operations, this feature limits its capacity. It is necessary to save the device energy as much as possible. Moreover, it is necessary to have mechanisms that allow dealing with the disconnection of the device when the battery is low or the signal is weak; **(iv)** *Ubiquitous Software Evolution Need*: to attend new market needs, the devices are constantly adapted, improved and evolved. Thus, it is necessary that the software, which is developed based on these devices, has mechanisms to allow its evolution; **(v)** *Different Devices*: the ubiquitous applications are based on different devices. Some of them are limited, such as a simple cellular. Others are powerful such as desktops and notebooks. Moreover, they have heterogeneous features in terms of memory and processing capacity; screen resolution, supported image types; and others; **(vi)** *Users' Mobility*: as the ubiquitous applications offer service/content anywhere and anytime, the users are in constant movement. In this context, it is important to consider technologies to deal with the intrinsic users' mobility through different smart-spaces; **(vii)** *Distributed Environment*: the ubiquitous applications are distributed and complex. They consider different intelligent environments that are distributed in the world.

4 OUR PROPOSAL

To deal with different ubiquitous challenges, such as the ones presented in Section 3, we propose the use of collaborative software agent communities based on agents' intentionality. The agents as a development paradigm have already demonstrated their benefits in various applications domains (Moulin and Chaib-Draa 1996). They can be used as natural abstractions to represent different users in ubiquitous contexts. Agents can take decisions to simplify the users' access to services, contents and peripheral devices. In this scenario and in order to emphasize the importance of the agents for ubiquitous computing development, some considerations that compose our proposal hypotheses are presented as follows: **(i)** *Software agents can asynchronously perform tasks*. It is desired in Ubiquitous Computing to deal with the challenge "Limited Battery of the Devices," presented in Section 3. Moreover, they allow parallel executions that reduce the user's waiting time to receive her/his requests; **(ii)** *Software agents significantly reduce the data traffic in the network*, because only the processing requests are transmitted by the network instead of several information packages. These requests are transferred with the agents; **(iii)** *Intentional software agents can improve the user's satisfaction* as they can be responsible for the content searching, analysis and adaptation according to the user and device profiles. Moreover, the agent can easily integrate the user to different smart-spaces based on the context analysis, and agent's beliefs, desires, and intentions. It is important to consider that the object orientation is based on the class abstraction that represents a group or a category. Thus, this abstraction can be used to satisfy a group but it is not appropriated to satisfy a specific member of this group. The agent abstraction can represent a specific member, a specific client, a specific device, a specific context. Thus, an agent can be used to satisfy a specific user according to her/his preferences, device' features, network specifications, and contract information instead of a group of users. Amongst other contributions, this approach increases the service personalization possibilities; **(iv)** *Software agents guarantee code portability* (Nakajima et al. 2003 *apud* Bergenti et al. 2004). It is important in Ubiquitous Computing as in this domain the smart-space distribution and the devices and network heterogeneity are intrinsic. The code must be portable, allowing the agent to move/circulate through the network without extra difficulties. Moreover, agents can perform their activities independently of the device operational system; **(v)** *Software agents, based on the agents' platforms protocols and resources, are uniforms*. They have pre-established communication protocols and life cycles. These features guarantee the ubiquitous system stability and avoid problems with heterogeneous networks; and **(vi)** *Software agents can be used to improve the security in ubiquitous applications*. As the ubiquitous applications are used in the people day-to-day activities, it is important that these applications are prepared to avoid malicious attacks. An agent can be responsible for the security. It can be modeled and implemented considering the security as a quality criterion.

4.1 Case Study Media Shop

The Media Shop project is based on a typically ubiquitous context, composed of: **(i)** heterogeneous devices (e.g. smartphone and palm); **(ii)** users in constant movement and with different preferences; **(iii)** several offered services and contents; and **(iv)** remote servers' integration need. The main objective is the media content download through different devices. The users' preferences are specified and stored in a knowledge base. They compose the user's profile. The agents consult this base to improve the service/content searching and consequentially the users' satisfaction.

The first solution to develop this project was based on: **(i)** the JADE platform (Bellifemine et al. 2007) to support the agents' implementation; **(ii)** a database to store the ubiquitous profiles; **(iii)** the Hibernate framework as the persistence layer; and **(iv)** the web-service for the implementation of the offered services such as the users' registration and the content download. This solution presents some deficiencies in the cognitive agents' development. According to our experiments the agents' implementation centered on their behavior showed itself not adequate to deal with the intentions of the users. The behavior abstraction is appropriate to represent user's behavior and the users also have intentions, principles and goals. Thus, we have the necessity to develop more intelligent, autonomous, flexible, and adaptable agents. We propose the use of a new technological support (second solution).

The second solution to develop this project was based on: **(i)** the JADE-LEAP platform (Caire 2003) to integrate different devices and smart-spaces; **(ii)** the JADEX framework (Pokahr and Braubach 2007) to support the development of cognitive agents centered on the BDI model; **(iii)** the JADEX-JADE Adapter to allow the integration between the JADE-LEAP and the JADEX; **(iv)** a database to store the ubiquitous profiles; **(v)** the Hibernate framework as the persistence layer; **(vi)** the use of fuzzy rules to improve the agents' cognition considering the non-functional ubiquitous requirements (e.g. user's satisfaction, performance, security, price); **(vii)** a collaborative and intentional MAS centered on the negotiation, adaptation and other agents' properties. We propose the use of intentional agents to represent the users, their devices and different ubiquitous contexts. We use the BDI model abstractions to improve the modeling and the development of cognitive agents' communities that perform actions, reason, learn and take decisions according to users' intentions, goals, beliefs, preferences, needs and other information capable of facilitating the service personalization in ubiquitous contexts.

4.2 Case Study DIUP Project

The DIUP project proposes the integration of the dental and the computing areas to improve the dental assistance in deprived communities. Different devices can allow patient registration, treatment follow-up; diagnosis, drug prescriptions; physician appointment management, image and registration searching, and other contributions. Our purpose is to perform these activities using an intentional MAS.

DIUP imposes various ubiquitous challenges such as: specific requirements analysis based on the dental domain; service omnipresence; content adaptation; different profiles and network bandwidth; servers overload; users' mobility; and context awareness. Moreover, there are a lot of stakeholders involved (e.g. dentists, patients and attendants); and various smart-spaces (e.g. deprived communities, hospitals and pharmacies)

Among the technologies used in this project, we can mention: **(i)** the i* framework (Yu 1998) to model the requirements; **(ii)** a development environment; **(iii)** the JADEX to implement cognitive agents; **(iv)** the JADE-LEAP to integrate different devices and the smart-spaces; **(v)** the JADEX-JADE Adapter to allow the communication between the JADE-LEAP and the JADEX; **(vi)** the BIGUS fuzzy logic library (Bigus and Bigus 2008) to improve the implementation of the agents' cognitive model (their reasoning and learning); and **(vii)** the intentional MAS centered on the negotiation, adaptation and other agents' properties.

4.3 Systematic Development

We documented the proposed technological support as a systematic development approach based on the RUP phases (*conceptual*, *elaboration*, *construction* and *transition*,) and the TROPOS disciplines. Thus, we organized the presentation of our proposal and its software engineering contributions according to the TROPOS disciplines: *Early Requirements*, *Late Requirements*, *Architectural Design*, *Detailed Design,* and *Implementation*. We also incorporated a discipline to present the testing proposal. We offer building blocks based on frameworks and models reuse for each discipline, as presented on Sections 4.3.1 to 4.3.4.

4.3.1 Perform the Early and the Late Requirements

The ubiquitous applications are very complex. Therefore, the requirements must carefully be elicited, modeled, and analyzed. Elicitation techniques combined with the stakeholders' interactions are extensile used by the authors and the collaborators to know the functional and non-functional requirements of both projects, Media Shop and DIUP. Moreover, we suggest the requirements modeling centered on distributed intentionality, using the i*framework models.

The early requirements discipline consists of the identification of the main application stakeholders, which are represented as actors in i* models. It is also necessary to elicit their goals based on the activities that they perform or will perform on the analyzed environment. In the late requirements discipline, we introduce the system as another actor. This actor is associated with other stakeholders, considering the dependencies among them. These dependencies represent the obligations of the system towards its environment and actors.

According to our experience using the i* framework to model our case studies, this modeling showed itself very adequate: **(i)** to deal with the agent's intentionality properties and the intentional relationships among different agents; **(ii)** to present descriptive models that consider the agent's strategic actions can be dynamic and unpredictable; **(iii)** to have a semi-formal component for concepts that are not prepared for formalization (e.g. user's satisfaction, security and performance).

Some advantages for using the i* framework are: the reduction of the quantity of diagrams, which are commonly used in other modeling proposals (Bauer and Odell 2005); the functional (Dubois 1989) and non-functional (e.g. softgoals) (Chung et al. 2000) requirements modeling centered on goal-orientation that increases the possibilities to implement traceability policies, and to model agents that dynamically decide which strategic plans will be performed according to the user's profiles, her/his device profile and contract profile. This latter advantage increases the possibilities to improve the user's service personalization, which is much desired in Ubiquitous Computing.

The i* uses two types of diagrams to model the application requirements: **(i)** the Strategic

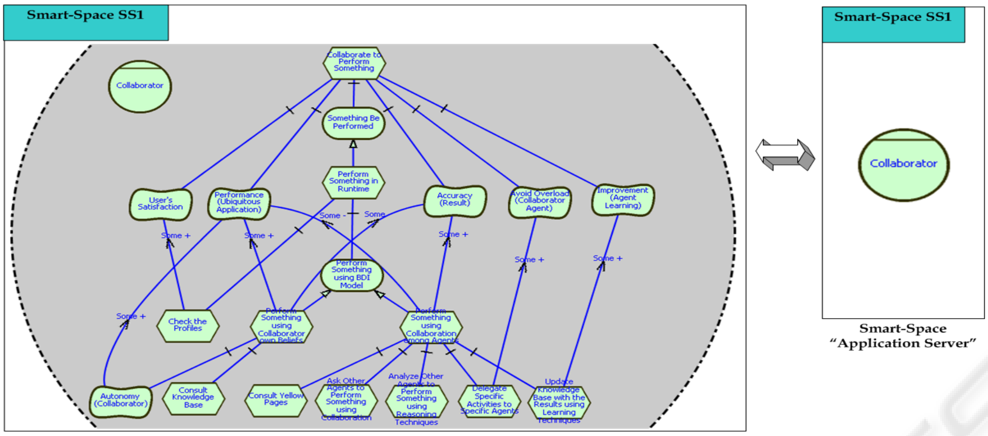

Figure 1: Smart-Space and Collaboration Representations.

Dependency Model (SD), which describes the dependences between actors/actors, actors/agents, and agents/agents in an organizational context; **(ii)** the Strategic Rationale Model, which describes the stakeholders’ interests, and how they can deal with different software specifications.

The i* framework offers some abstractions for the goal-oriented modeling, such as actor, agent, task, goal, softgoal, resources, means-end, and task-dependency. Details about these abstractions can be obtained in (Yu 1997). These abstractions are not sufficient to represent all the concerns of Ubiquitous Computing for many reasons, such as: **(i)** i* does not consider a specific abstraction to represent different smart-spaces; **(ii)** i* does not consider a specific abstraction to represent the intrinsic agent’s mobility. Thus, our approach suggests some complementary and specific abstractions for agent-oriented ubiquitous applications modeling: **(a)** *Mobile Autonomous Entity*: is an autonomous entity (reactive or cognitive agent) that is capable to move from a smart-space to another. We suggest to use the traditional i* agent abstraction combined to different tags or stereotypes to classify the agent as fixed or mobile. In the case of the mobile autonomous entity it is classified as the *mobile* stereotype; **(b)** *Smart-Spaces*: are intelligent ubiquitous environments, in which the actors and the agents perform and deliberate plans to achieve their goals. We suggest the box representation (see Figure 1), which delimits the smart-space that is composed by an agent or an agents’ community; **(c)** *Collaboration*: is desired that the agents in ubiquitous context cooperate with each other to achieve the specified purposes. We suggest the use of a set of goals, tasks and softgoals combined to task-decomposition to represent the collaboration. In order to simplify this representation, it is recommendable the use of the *capability* representation. This representation encapsulates all the collaboration process. Other relationships (e.g. negotiation, concurrency, and adaptation) between the agents can also be represented using *capability*. The collaboration between two agents is also illustrated in Figure 1.

Some other abstractions are presented on our technical report (Serrano et al. 2008[d]), submitted for IEEE Transactions. Additionally, we are identifying the different ubiquitous non-functional requirements (e.g. user’s satisfaction, performance, security, and service omnipresence), the impact among them, and ways for operationalizing them using intentional agents. Details are presented on Section 4.3.2.

4.3.2 Perform the Architectural and the Detailed Design

The architectural design considers some new actors/agents and the subgoals/subtasks, which are associated with these actors/agents and the system in development. In the detailed design, we detail the system actors/agents. Thus, the communication between them and the respective protocol that will be used to orient it are established. The models constructed in the early and late requirements are the main entries of the design disciplines.

In the design disciplines, our approach suggests some important models, framework and middleware based on the literature and the experimental results, obtained in the Software Engineering Laboratory at PUC-Rio. Some models and frameworks are: the JADE-LEAP execution modes; the JADEX BDI model; the Intentional Framework for Content Adaptation in Ubiquitous Computing (IFCAUC) (Serrano et al. 2008[c]); the incremental spiral model; and the Softgoal Interdependency Graph (SIG) (Chung et al. 2000). Our purpose is to assist the software engineer to systematically develop ubiquitous software based on reuse. The *JADE-LEAP platform* is basically the JADE used to run in mobile devices as Personal Java and MIDP devices. This platform offers two basic execution modes, called *standalone* and *split* modes. The standalone mode allows integrating the platform and the Personal Java devices, which are capable of running the platform container. The split mode allows integrating the platform and the MIDP devices. In these devices, as they are limited in memory and processing capacities, the JADE-LEAP provides a mechanism to allow the devices to share resources with a powerful machine. In other words, the limited devices can run only the "light" part of the container, called **front-end**, and the "strong" part, called **back-end**, can run in the powerful machine. For the limited devices, this integration process is invisible. Thus, using these two modes it is possible to integrate both, limited and powerful devices, to the agents' platform.

The *JADEX BDI model* is important to implement the cognitive agents. The BDI model considers that the agent is based on its beliefs, desires, and intentions. This model uses the abstractions: **(i)** agents, which represent entities that perform the actions delegated by the users; **(ii)** beliefs, which represent information, knowledge and physical object of the real world that can be exchanged among the agents; **(iii)** desires, which represent the goals that will be achieved by the agents; and **(iv)** intentions, which represent a set of actions/tasks to achieve the specified goals. This set of actions/tasks is called plan.

The *IFCAUC* is a framework, developed by the authors, based on an intentional MAS to perform activities involved in the content adaptation process. This model recommends, among other technologies, needs and concepts: **(i)** the use of different strategies centered on agents' intentionality to perform the content adaptation according to the ubiquitous profiles and in order to guarantee users' satisfaction; **(ii)** the use of different profiles as a knowledge base to store the users' preferences, the access devices' features, the contract and service information, and the network specifications; **(iii)** the use of a MAS centered on collaboration, negotiation, concurrency that is improved by reasoning and learning techniques and the agents' mobility; and **(iv)** the use of a dedicated server to perform the adaptation and to avoid the application server overload.

The *spiral model*, which is used in traditional application development, is also suggested in our approach. However, we incorporated the concept of the incremental software construction in this model. Thus, in our incremental spiral model, we based the ubiquitous application development on the generation of various versions, each more improved than the last. This incremental principle allows the evolution and the adaptability of the ubiquitous software development as errors on the implementation or on the design can previously be identified and corrected before the software deployment. Figure 2 shows this model.

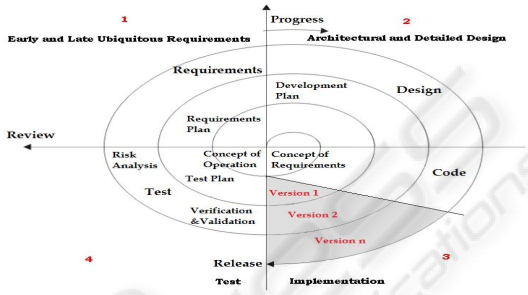

Figure 2: Incremental Spiral Model.

The *Softgoal Interdependency Graph (SIG)* is used to formalize the ubiquitous non-functional requirements (NFRs), the impacts among them, and ways to operationalize them using an intentional agent-oriented approach. The main objective is to provide an adequate support for software engineers to develop ubiquitous application from the requirements elicitation to code, considering the non-functional requirements. Software engineers frequently forget these requirements. Thus, as our approach offers a way to deal with these requirements from the modeling to code, it considerably increases the relevance of our systematic development. Figure 3 illustrates some operationalization for the Satisfaction[User] and Invisibility[Process Complexity] NFRs.

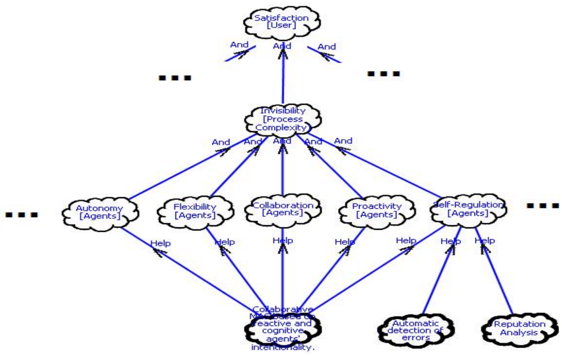

Figure 3: SIG for Satisfaction and Invisibility.

Additionally, another interesting contribution of our proposal consists of the associations between the i*, the BDI, and our approach abstractions (IUSDT). Figure 4 illustrates some of these associations.

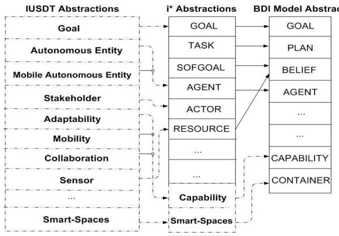

Figure 4: Association among Abstractions.

Furthermore, we are developing an ontology for context aware ubiquitous application to improve and to standardize the agents' knowledge base. We also incorporated, for example, the smart-space and the profiles abstractions as terms of our ontology. Figure 5 shows the hierarchy of our ontology. Other models are presented in (Serrano et al. 2008[a] and 2008[b].)

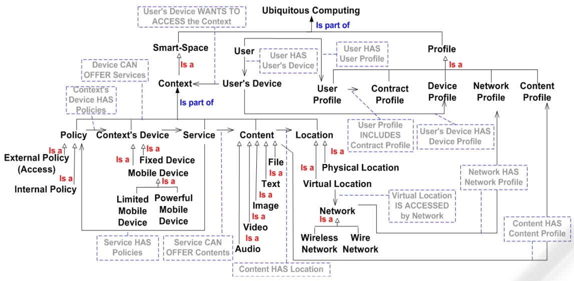

Figure 5: Ontology Hierarchy.

4.3.3 Perform the Implementation

The implementation discipline depends on the details specified in the design disciplines. The TROPOS approach suggests an association between the detailed design and the implementation template. The code is added to the template using agent-oriented programming language supported by a platform (e.g. JADE-LEAP) or a framework (e.g. JADEX). The models of the previous disciplines are the main entries in the implementation.

The support offered by the proposed approach is centered on framework and middleware reuse. Therefore, the JADE-LEAP, the JADEX, the Hibernate, the IFCAUC, and other technological supports are used in the ubiquitous software implementation. This technological set has the responsibility to offer an adequate infra-structure to solve the main ubiquitous challenges, improving the software engineering work in the integration of different devices and smart-spaces; in the users' mobility, in the cognitive agents development based on ubiquitous profiles; in the communication, registration and creation of the agents; in the storage and recovery of ubiquitous profiles information; in the adaptation of contents, which are requested in different access devices anywhere and anytime; and in the context analysis to facilitate the user's integration in different ubiquitous smart-spaces.

Some frameworks (e.g. JADEX and Hibernate) are under development by important and renowned Software Engineering research groups. Others, such as the IFCAUC, are being developed by the authors to improve the development of intentional ubiquitous application considering specific ubiquitous concerns (e.g. content adaptation, ubiquitous profiles need and context awareness). Implementation details can also be obtained in (Serrano et al. 2008[a] and 2008[b] and 2008[c].) As follows, we briefly describe some ubiquitous application implementations based on the case studies presented on Sections 4.1 and 4.2, which are developed in the Software Engineering Laboratory at PUC-Rio as experimental researches of our group.

(1) Framework for Media Content Download using Mobile Devices centered on the Agents' Behavior and the Web Service Technology: we develop a framework for download media contents (e.g. music and video). The users requested and received the contents using their own access devices, which could be limited or not. The main technologies involved in this implementation were: JADE, web-service, MAS, user and device profiles. The agents, according to the implementation model suggested by the JADE, were based on their behavior. This model, based on our experimental results, did not allow improving the agents' cognition to quite represent the human intentions in order to guarantee the user's satisfaction, and the agents' autonomy, adaptability and flexibility.

(2) Ubiquitous Applications instantiated using the Framework **(1)**: we instantiate the previous framework to develop three ubiquitous applications. The first application instantiates the framework to download the contents using a desktop and a web interface. Our purpose was testing different powerful devices with several users' requests. The desktops are integrated by the wire network. The second application instantiates the framework to download the contents using mobile devices with limited memory and processing capacities. The third application instantiates the framework to register the offered contents and to download the requested content. According to the performance testing, the framework agents are very efficient. However, with regard to the user's satisfaction, as the cognitive model is very limited, the results are medium. We improved this framework by using emergent technologies, as the ones presented in the next framework implementation.

(3) Framework for Media Content Download centered on the Agents' Intentionality: in order to improve the cognitive agents' model, their autonomy, flexibility and adaptability, we use emergent technologies to implement the MAS. Some of these technologies are: the JADEX and the BDI model; the JADE-LEAP to integrate different devices and smart-spaces; ubiquitous profiles (user, device, contract, network, and content profiles) to store and to recover specific information to achieve the user's satisfaction; the Hibernate as the persistence layer; fuzzy rules to improve the agents' reasoning. The first structure layer of our Media Content Framework represents the interface, in this case the user and her/his access device. The second layer is more complex and represents the intentional MAS. The technological support that comprises this layer is the JADE-LEAP, the JADEX, the JADEX-JADE Adapter, and the BDI model abstractions. The third layer represents the knowledge base, which is used by the agents to perform the tasks in order to achieve the goals. The knowledge base is composed of the agent's beliefs, the fuzzy rules based on non-functional ubiquitous requirements, and the ubiquitous profiles. The agents consult this knowledge base, improve their reasoning, and dynamically decide which strategy is the best one to guarantee the user's satisfaction and the device compatibility. In the performance testing, the results are excellent. The difference in relation to the first proposed implementation (agents based on behavior) was in the user's satisfaction. Using intentionality and a solid knowledge base, the agents were more cognitive and it also improves their strategies, which guarantee and increase the user's satisfaction.

(4) Framework for Content Adaptation in Ubiquitous Computing centered on Agents' Intentionality (IFCAUC): when we defined the appropriate technologies, we developed a framework to deal with a specific concern of ubiquitous applications, the content adaptation. We also developed a conceptual model to improve the design and the implementation of the IFCAUC. This conceptual model is detailed in (Serrano et al. 2008[c]). Different strategies to improve the agents' performance and to guarantee the user's satisfaction were used in the agents' implementation. We structured these strategies in a pyramid representation (see Figure 6), in which the agents' first goal is the user's satisfaction. If it is not possible to achieve the user's satisfaction, the agents perform activities to guarantee device compatibility. The agents' intentionality is improved by using the BDI model. The agents' knowledge bases are dynamically updated, which guarantee more realism in the agents' decisions. Moreover, we use mobile agents to perform some complex content adaptations on a dedicate server. It avoids the application server overload. We use the capabilities offered by the JADEX to register and deregister the agents in the Yellow Pages; to control the agents' life cycle in the White Pages; and to improve the agents' mobility. These programming practices standardized the implementation and facilitated the IFCAUC instantiation.

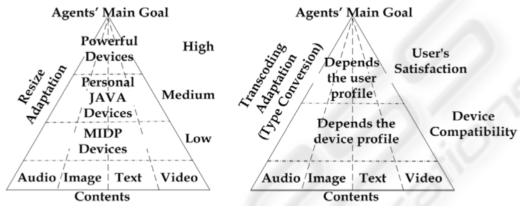

Figure 6: Strategies Pyramids for Content Adaptation.

(5) Framework for Context Awareness in Ubiquitous Computing centered on Agents' Intentionality: we developed a context awareness framework to allow the analysis of different contexts and to facilitate the integration of users and smart-spaces that offer services, and peripheral access. Our idea is to use agents to represent the users' interests and to negotiate with other agents in order to integrate the user in the smart-spaces.

(6) Applications using Physical Sensors and Intentional Multi-Agent Systems: we developed some applications to test different methods of communication between the users and different contexts or the users and different devices. We explored physical sensors such as touch sensor, presence sensor, sound sensor and light sensor.

4.3.4 Perform the Test

This discipline is dependent on the previous disciplines. Our approach, as we are following the TROPOS methodology, suggests the requirements-driven ubiquitous development. Therefore, we use the SD and SR models to previously select which plans will be executed in order to achieve the specified goals. In this scenario, it is important to consider, for example: the ubiquitous profiles; the available technologies; and the non-functional requirements. Our purpose, based on the requirements verifiability in high abstraction level, is to stimulate the ubiquitous software development centered on quality and in agreement with the users' interests. Our proposal, combined with the incremental development, can reduce the test costs and avoid the need for rigorous validation tests.

In order to validate our approach, we performed some experimental tests with the user participation. The results are promising and confirm the pertinence of our approach. The first test is performed in the IFCAUC. We model this framework using agents and goal-oriented approach. Thus, we verify, by using a modeling analysis, if each agent's decision was according to the ubiquitous requirements. Additionally, some stakeholders were invited to participate in the evaluations. Our purpose was the investigation of users' satisfaction in the DIUP application, instantiated using the IFCAUC. Each stakeholder specifies her/his preferences, which were stored in the user profile. Moreover, we requested that each stakeholder used her/his own access devices. The devices' features are registered in the device profile. The dentists requested some content (e.g. x-ray image) using their devices. The DIUP application analyzed the requests and returned the contents, which were adapted according to the user and device profiles. Therefore, the stakeholders declared how satisfied they were with the received content. The main metrics we used are: **(i)** the answer time; **(ii)** the service/content quality; **(iii)** the service/content cost; **(iv)** the security to obtain the requested content; and **(v)** the relevancy of the offered content. *Results*: 80% of the users were *Completely Satisfied*, 10% were *Completely Satisfied/Very Satisfied*, and 10% were *Very Satisfied*.

These results demonstrate high indices of users' satisfaction in relation to the quality and the relevancy of the requested contents. The stakeholders also emphasized the DIUP application performance in quickly answering the requests. These good results only strengthens the fact of our proposal is appropriate to ubiquitous software development as it considers different ubiquitous profiles, users' satisfaction, service omnipresence, agents' performance, and other requirements that are commonly found in ubiquitous context.

We also analyzed the agents' performance using a log agent, which is responsible to determine the answer time. This time started when the content was requested and finished when the adapted content was sent to the user in her/his device. The contents had different resolution and adaptation needs. The results were optimistic, and raised the authors' confidence in the proposed approach.

Furthermore, our approach suggests an automated non-functional (NF) testing using MAS. We are developing an intentional agent-oriented framework for NF testing in Ubiquitous Computing. We will analyze, for example, the ubiquitous application performance and security using stress and trustworthiness testing, respectively. We intend that the agent community generates a report according to different NF testing.

4.4 Summarizing Our Systematic Development Proposal

It is important to consider that the emphasis of the *Conceptual Phase* in our proposal is in the Early and the Late Requirements disciplines, in which we are concerned about the stakeholders' interests, the organization's policies and the ubiquitous system that will be implemented. In the *Elaboration Phase* our proposal suggest that the software engineer dedicates her/his time on the Architectural and Detailed Design disciplines, in which we are establishing the architectural model and the technological support that will be used in the ubiquitous software development. In the *Construction Phase* we are particularly motivated by the challenges imposed by the Implementation disciplines, in which we will apply all the conceptual and technological support, specified in the previous phases. In the *Transition Phase* we concentrate our attention on testing the ubiquitous software, using the users' participation and considering the users' feedback as a parameter to classify the ubiquitous software development as a success or not. Figure 7 illustrates these disciplines and phases.

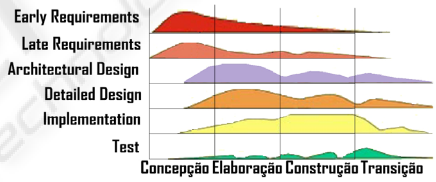

Figure 7: Summarizing Our Systematic Development.

As briefly showed in previous sections, our systematic development is based on building blocks for each discipline. For example, the JADE-LEAP is used to deal with the integration of different devices. Thus, this platform is part of our *Device Integration Building Block*. In other words, we propose a systematic development using building blocks based on platforms, frameworks and models reuse.

5 CONCLUSIONS

Ubiquitous Computing encompasses: **(i)** content adaptation needs; **(ii)** integration of devices and smart-spaces considering the users' mobility and the devices' heterogeneity; **(iii)** adaptability capacity of ubiquitous applications in a world in constant evolution; and others. In order to deal with these challenges, our proposal investigates traditional and emergent technologies centered on intentional MAS to improve the ubiquitous software development.

The main advantages of MAS include: autonomy, reactivity, ability to decide without human interaction; adaptability, mobility, interactivity, and flexibility. Intentional agents have all the advantages presented above and also represent a natural metaphor for human actions as they can decide centered on their beliefs, desires and intentions. Thus, we argue that the use of an intentional MAS seems to be adequate in the ubiquitous applications development.

Another contribution of our approach proposes an incremental development for ubiquitous applications, which is based on reuse. We combined traditional and emergent technologies to specifically support the ubiquitous software development, which is improved using an ontology and fuzzy logic.

As further work, we intend to improve our approach, incorporating technologies to deal with the ubiquitous concerns using MAS. The agents' cognition can be based on special issues on learning techniques for compositional reasoning (Giannakopoulou and Păsăreanu 2008) (Upal and Rogers 2005). It can be a way to discover a new basis for building intelligent agents. Moreover, intentional agents can also be considered as planning systems (Wilensky 1983). We intend to investigate this research area in Ubiquitous Computing.

REFERENCES

Bauer, B., Odell, J., 2005. *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.

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

Bergenti F., Gleizes, Marie-Pierre, Zambonelli, F., 2004. *Methodologies and Software Engineering for Agent Systems*. The Agent-Oriented Software Engineering Handbook, Kluwer Academic, ISBN: 978-1-4020-8057-9, Vol. 11, 506 pages.

Bigus, J., Bigus, J., 2001. *Constructing Intelligent Agents Using Java*. 2[a] Ed. John Wiley & Sons, ISBN 978-0471396017, 432 p., March.

Caire, G., 2003. *LEAP User Guide*. Copyright (C) TILAB, LEAP Version 3.1, December.

Chung, L. K., Nixon, B. A., Yu, E., Mylopoulos, J., 2000. *Non-Functional Requirements in Software Engineering*. Kluwer Academic, ISBN 978-0-7923-8666-7, Vol.5, 472 pages.

Dubois, E. A., 1989. *Logic of Action for Supporting Goal-Oriented Elaborations of Requirements*. 5[th] Int. Workshop on Soft. Specif. and Design, ACM SIGSOFT, Vol. 14, Issue 3, pp. 160-168, May.

Giannakopoulou, D., Păsăreanu, C. S., 2008. *Special issue on learning techniques for compositional reasoning*. Journal on Formal Methods in System Design, Springer Netherlands, ISSN: 0925-9856, Vol. 32, Number 3, pp. 173-174, May.

Giunchiglia, F., Mylopoulos, J., Perini, A., 2002. *The Tropos Software Development Methodology: Processes, Models and Diagrams*. 1[a] Int. Conference on AAMAS, ISBN: 1-58113-480-0, pp. 35-36, Italy.

Hightower, J., Borriello, G., 2001. *Location systems for ubiquitous computing. Computer*. IEEE, Vol. 34, Number 8, pp.57-66, August.

Kruchten, P., 2003. *The Rational Unified Process: An Introduction*. Addison-Wesley, ISBN: 0321197704, 3[a] Ed, 320 pages, USA.

Moulin, B., Chaib-Draa, B., 1996. *An Overview of Distributed Artificial Intelligence*. Chapter 1, Book Foundations of Distributed Artificial Intelligence, ISBN: 978-0-471-00675-6, pp.3-55.

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

Pokahr, A., Braubach. L., 2007. *JADEX User Guide*. Distributed Systems Group, University of Hamburg, Germany, Release 0.96, June.

Ravindran, G., Jaseemudin, M., Rayhan, A. A., 2002. *Management Framework for Service Personalization*. Management of Multimedia on the Internet, Lecture Notes in Computer Science, Springer Berlin, ISBN 978-3-540-44271-4, pp. 276-288, January.

Saha, D., Mukherjee, A., 2003. *Pervasive Computing: A Paradigm for 21st Century*. IEEE Computer.

Serrano, Milene, Serrano, Maurício, Lucena, C. J. P., 2008[a]. *Engineering Multi-Agent Systems for Design and Implementation of Ubiquitous Computing*. Monograph 10/08, PUC-Rio, 25 pages, March.

Serrano, Milene, Serrano, Maurício, Lucena, C. J. P., Leite, J. C. S. P., 2008[b]. *Modeling Ubiquitous Applications based on Agent's Intentionality and Multi-Agent Systems*. Monograph 13/08, PUC-Rio, 25 pages, March.

Serrano, Milene, Serrano, Maurício, Lucena, C. J. P., 2008[c]. *Framework for Content Adaptation in Ubiquitous Computing Centered on Agents Intentionality and Collaborative MAS*. 4[th] Workshop on Soft. Eng. for Agent-Oriented Systems (SEAS), 12 pages.

Serrano, Milene, Lucena, C. J. P., 2008[d]. *Intentional Ubiquitous Software Development Technologies (IUSDT)*. Submitted to IEEE Transaction - Journal on Mobile Computing, 35 pages, October.

Upal, M. A., Rogers, S., 2005. *Learning to Improve Reasoning*. Computational Intelligence, Vol. 21, Number 4, pp. 333-335.

Wilensky, R., 1983. *Planning and Understanding: A Computational Approach to Human Reasoning*. Advanced Book Program, Addison-Wesley, MA.

Yu, E. S. K., 1997. *Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering*. 3rd IEEE International Symposium on Requirements Engineering, pp. 226-235, Washington, USA.