Applying Regulation to Ubiquitous Computing Environments [*]

José Viterbo F.[1]**, Carolina Felicíssimo**[1]**, Jean-Pierre Briot**[2]**,**
Markus Endler[1]**, Carlos Lucena**[1]

[1]Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro (PUC-RJ)
R. Marquês de São Vicente, 225
22453-900, Rio de Janeiro, Brazil

[2]Laboratoire d'Informatique de Paris 6 (LIP6)
Université Paris 6 - CNRS
8 rue du Capitaine Scott
75015 Paris, France

`{viterbo,carol}@inf.puc-rio.br, briot@poleia.lip6.fr,`

`{endler,lucena}@inf.puc-rio.br`

Abstract. *Ubiquitous computing systems may be considered as typical open systems, where due to the mobility of devices, heterogeneous and previously unknown entities may come to interact spontaneously. The standard context-aware ubiquitous systems are mainly concerned with topological space and resources availability. But a topological space is also populated with agents, human or artificial, that act and interact socially. This paper describes our proposal for integrating a context-aware approach with a social norms regulation approach. More precisely, we discuss a first prototype integration of MoCA architecture with the social norms regulation architecture DynaCROM. A small scenario involving professors and students collaborating within and across several universities is used as an example and as a first test case for our approach.*

1. Introduction

In the vision of ubiquitous computing, computer systems will seamlessly integrate into our everyday lives, providing services and information anytime and anywhere [Weiser 1991]. Ubiquitous computing aim at exploiting the full range of sensors and networks available to transparently provide computational services, regardless of time and the end user's location. Compared to traditional distributed systems, ubiquitous computing systems feature increased dynamism and heterogeneity [Soldatos et al. 2006]. The underlying ubiquitous computing infrastructures are more complex and bring into the foreground issues such as user mobility, disconnection, dynamic introduction and removal of devices, heterogeneous network connections, as well as the need to integrate the physical environment with the computing infrastructure [Kindberg and Fox 2002].

In fact, the support to user mobility is one of the most important challenges in the design of ubiquitous systems. Users should be able to take full advantage of the local capabilities and resources within a dynamic environment, where users and devices

---

[*] Work partially funded by projects CNPq 55.2068/02-2 (ESSMA) and 479824/04-5 (Ed. Universal)

frequently enter and leave that environment, and resources (e.g. available bandwidth) change [Sousa and Garlan 2002]. The environment should support automated adaptability using policies that govern the behaviour and interactions among users and applications [Harroud et al. 2004].

The standard context-aware approach to design and construct ubiquitous systems usually focuses on two dimensions: the topological space (e.g., to represent rooms, locations) and the resources (e.g., when battery is low or when network is overloaded). But space is also a social scenario for the interaction of agents — human and artificial —, where social rules should be applied and influence the interaction process. Thus, our main goal is to extend common support for ubiquitous computing with means of managing the social context, through representation and management of social norms.

Ubiquitous systems, as typical open systems, may contain dynamically interacting components engaging in complex coordination protocols. In such kind of systems, entities, human or artificial, communicate with each other as a means of interacting either to cooperate or to compete. Since there is no prior knowledge of which agents will enter in the environment and interact with all others, it is not feasible to implement all possible interaction behaviours into the agents. For this reason, a law enforcement approach should separate the rules from the entitie's implementation, making them explicit [Paes et al. 2004].

This work proposes a system that applies regulatory mechanisms to coordinate the interaction among heterogeneous entities in an ubiquitous computing scenario. The next section discusses the current ubiquitous support to applications and users. Section 3 presents our scenario. In Section 4, we discuss our approach. Section 5 presents the architecture of our system. In Section 6, we present a case study scenario. In Section 7 we show the implementation status. Finally, Section 8 brings the conclusions about the proposed system and future work.

2. Related Work

Several proposals have already been presented to support service providing and applications in ubiquitous systems [Johanson et al. 2002, Sousa and Garlan 2002, Rudolph 2001, Soldatos et al. 2006], but up to now, most of them have been concerned with the topological aspects of resources distribution. Interactive Workspaces [Johanson et al. 2002] project, for instance, concentrates in the human interaction with devices and large high resolution displays. Services and devices indicate their presence posting short duration events in an event queue related to a given physical space. Any entity may query that queue to find out available services inside an area of interest, based only on the service description.

Another system, the Oxygen [Rudolph 2001] project, foresees a future in which computational power will be freely available anywhere, as oxygen in the air we breathe [Saha and Mukherjee 2003]. The project proposes a user-centric support for ubiquitous applications, emphasizing specially the automatic and personalized access to information, adapting applications to users preferences and necessities as she moves through different spaces [Soldatos et al. 2006].

We may also cite the Aura [Sousa and Garlan 2002] project, which is based on the idea of a personal information aura that spreads through portable devices and fixed computers around a user [Saha and Mukherjee 2003]. Aura provides a software architecture that monitors an application and guides its dynamic changes, thus providing opportunities to adapting it to variable resources, users mobility, changes in users needs and system failures [Soldatos et al. 2006]. In practice, when a user moves through different spaces, the tasks she is undertaking are carried with her with the use of local resources.

Finally, ACAI [Harroud et al. 2004] presented an approach for supporting mobile ubiquitous applications based on a framework that provides necessary features and services to facilitate the building of context-aware mobile applications. The framework uses semantically modeled policies to adapt the application behaviour dynamically based on context information. Policies are represented semantically to help achieve common undertanding across different domains and to allow inference for automating the process of generating implicit policies.

3. Scenario

As a typical scenario to exemplify our approach we consider two universities in two different countries, for instance, PUC-Rio, in Brazil, and LIP6, in France. We assume that both organizations have a location service capable of determining the position of a user — who carries a mobile device — inside the organization. This information is given in terms of symbolic location, which associates a name with each of the smallest areas distinguishable by the location service. Those atomic spaces may be classrooms, laboratories, seminar rooms, offices, corridors, etc. Although each of these spaces is distinct and has an exclusive geographical position, some of them have identical functions.

People who populate those environments are assumed to be associated with these organizations, e.g. as teachers, students or administrative staff members, or may be just passing by. We further assume that the two organizations have some sort of cooperation, so that a member of one institution may be a temporary visitor at the other institution.

Each person in this scenario carries a mobile device connected to a wireless network and where he runs applications that access different available services, some of which may be subject to location-based regulation. For instance, when a lecturer enters a classroom of his university, he can use a specific client application to access a datashow within that same place, but if he is in a classroom of another university he can't. A student in that same university may interact with his teacher and other students through a chat client when he is inside a classroom, but as soon as he leaves the room the service is turned off. These functionalities would be useful to implement an application to automate classrooms, for instance. On the other hand, we can think of regulation also to impose automatic restrictions to applications, according to the specific norms of a institution. For example, a student may receive some messages as he walks along the university corridors, but when he enters the office of a lecturer his messages will be cached to be delivered after he leaves. These are some examples in which services are provided according to norms that depend on the semantic location of a user and his relationship with that institution.

4. Our Approach

In this work, we assume that an open multi-agent system (MAS) is a system that puts together a set of heterogeneous agents whose actions may deviate from the expected behavior in a context. A regulated (or normative) open MAS provides norms that support regulation over the agent's actions. In this scope, a norm model is used as means to formalize norm regulations. In addition, we think that the model must also provide a rule support mechanism to assist the regulation of agents during system execution, and that this mechanism should be flexible, easy to operate, and permit that norms be created, deleted and modified at runtime [Felicíssimo et al. 2006a].

Note that actual regulation by agents can take two forms. In the first approach, agents use the information about current norms (actual norms associated to an agent may dynamically change depending on its location, role, etc.) as an additional criterium to decide what to do (action selection). An example is the BOID architecture, extending a BDI (Belief, Desire, Intention) architecture with Obligations [Broersen et al. 2001]. In the second approach, agents are considered as black boxes and an external mediation mechanism enforces that interactions or actions follow the norms. An example is the XMLaw architecture for enforcing agent interactions through a mediator which blocks prohibited messages, and may generate new messages in case of obligations [Paes et al. 2004]. Our proposal is in fact generic and may be integrated with both approaches. Meanwhile in current implementation and current scenario, we used the first approach. Integration with the second approach will also be considered in the future.[1]

We believe that in the same way a regulation mechanism may be used to control the interactions among entities in a open system, it could also be used to control the interactions among entities of an ubiquitous system. In such case, users carrying mobile devices can walk through different environments, and in each of them the software entities running at her devices will have to interact with different entities, each responsible for a given set of services restricted to the current environment.

To know the entities that populate a environment at each moment, this system will need to receive context information concerning the devices involved, that is, the location of each device used in the system. MoCA (Mobile Collaboration Architecture) [Rubinsztejn et al. 2004] is an architecture that supports the development of context-aware applications for mobile computing. Among other context information that it provides, MoCA delivers the location information of a mobile device inferred from the signals received from 802.11 network access points. This makes it appropriate to be used as a location information provider for indoors environments.

As our norm model we use the DynaCROM [Felicíssimo et al. 2006a] approach. This model considers four general scopes of norms: Environment, Organization, Role and Interaction. Environment norms are applied to all agents from a regulated environment. Organization norms are applied to all agents from a regulated organization. Role norms are applied to all agents playing a regulated role, and finally, Interaction norms are applied to all agents involved in a regulated interaction. These regulatory scopes have their semantic explicitly represented in an generic ontology [Felicíssimo and Lucena 2005], in which they are represented by the basic classes Environment, Organization and Role. For each case, the generic ontology has to be extended and instantiated, with the creation of specific subclasses, instances and properties related to each class. For example, we can think of University and High School as subclasses of Organization, and PUC-Rio as an

---

[1] "Do we change the structure of the corporation? — And it is obviously in the long term — Or do we try to regulate its behavior through, say, environmental laws, labour laws, human rights laws? I think we need to be doing both."[Bakan 2003]

instance of the subclass University. Or Country and City as subclasses of Environment and Brazil as an instance of the subclass Country.

In our approach, while the Environment scope has a topological semantics, Organization, Role and Interaction scopes account for the social aspect, all at the same level, i.e. there is no preestablished priority for the enforcement of norms. The Environment describes physical spaces, places such as buildings or rooms, but specialized classes may be associated to specific spaces with common norms in different organizations, a classroom, for instance. The Organization describes some social structure or institution, like a university, a company, or a department. As these scopes are orthogonal, they may be intertwined, i.e, the relationships between subclasses and instances of the basic classes may be defined freely. For example, we may have a classroom 511 (Environment instance) at PUC-Rio (Organization instance), in Brazil (Environment instance), and a classroom 27 (Environment instance) at LIP6 (Organization instance), in France (Environment instance). A Role may have specialized classes that indicate some social role valid across organizations and environments. For example, Marie, who is a student (Role instance) from LIP6, may be visiting PUC-Rio and attending a meeting at classroom 511. Each of these context instances may have some norms associated to them. Rules of composition describe how these norms may be considered to infere a final set of norms that are applicable in a given situation.

5. Architecture

We present a MAS in which we use DynaCROM, a regulatory mechanism that supports the dynamic composition of norms based on rules, to guide and coordinate the interaction among the users' agents and the service providers. For handling the context information necessary to implement this system, we used MoCA architecture and MoCA/MAX extension. Our proposed architecture is detailed in the following subsessions.

5.1. The MoCA architecture

The MoCA architecture [Rubinsztejn et al. 2004] provides support for developing and executing distributed context-aware applications, particularly those that comprise mobile devices interconnected through wireless infrastructured LANs (802.11b/g). The services provided by MoCA support the collection, distribution and processing of context information acquired directly from the mobile devices or inferred through context services. In addition, MoCA offers a set of APIs to support the design and implementation of context-aware applications that use the MoCA services.

In Figure 1 we see the three main components of MoCA. A service called Monitor must execute on each mobile device to collect raw context data. The Context Information Service (CIS) is responsible for collecting, storing and processing this raw data, and the Location Inference Service (LIS) is responsible for inferring the location of a mobile device from the information about RF signal patterns received from reference points [Nascimento et al. 2006].

Besides context management, CIS and LIS also implement APIs to access context information from applications, allowing clients to register their interest in specific context states (involving one or several context variables) modeled as logical expressions, and to be asynchronously notified whenever the corresponding context-expression is satisfied [Sacramento et al. 2004]. This functionality reduces the cost to build client applications, since they don't have to manage the context information delivery.

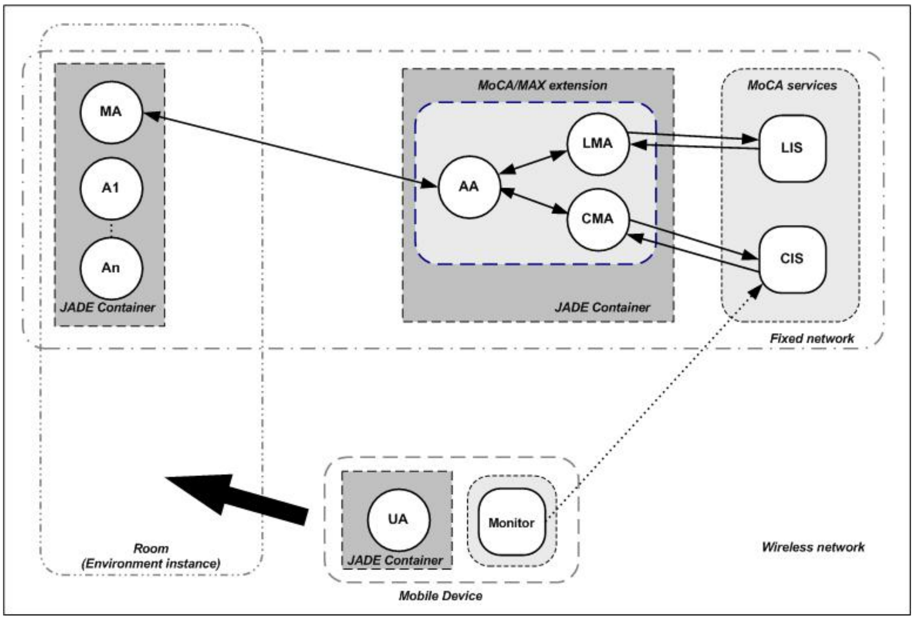

Figure 1. Architecture of the proposed system

5.2. JADE and MoCA/MAX

For the development and deployment of multi-agent systems, an integrated and distributed control environment is necessary to provide the basic services and functionalities defined by the software agents paradigm. JADE (Java Agent DEvelopment Framework) is one of the most popular available middlewares that provide such an environment [Bellifemine et al. 2001]. A JADE platform comprises one or more agent containers executing on one or more hosts interconnected by a data network. Each container provides the proper environment for the creation and execution of one or more agents.

To be able to create context-aware applications using JADE and MoCA in a simpler way, we developed a MoCA extension called MoCA/MAX (MoCA/Multi-Agent eXtension). The middleware resulting from the conjunction of JADE and MoCA/MAX facilitates the development of applications built upon the JADE framework and required to use context information provided by MoCA. Agents interested in context information communicate with MoCA/MAX, to query or subscribe for context information exchanging ACL-messages. Figure 1 shows the main agents of MoCA/MAX: the Advertisement Agent (AA), the Context Management Agent (CMA) and the Location Management Agent (LMA). The AA acts as an interface between CMA/LMA and the agents requiring access to context information. This agent represents a unique and well-known address within the platform to where context consumers must send messages asking for information. The AA forwards to CMA the messages related to requests for a device's computational context managed by CIS, and forwards to LMA the messages about a device's location information inferred by LIS. The reply messages from CMA and LMA are also first sent to AA, which then forwards them to the requesting agents.

5.3. DynaCROM

We have extended the DynaCROM ontology [Felicíssimo et al. 2006a] to comprise new relations among the basic classes. As the first three regulatory scopes, Environment, Organization and Role, are non-hierarquical, we proposed the extended ontology depicted in Figure 2. In our proposal, an Environment may have different dimensions, and the smallest dimension defines a active space where a set of resources (devices and services) is controled by a monitor agent that will regulate the interaction among these resources and the agents representing the users that come into this active space. A small size instance of an Environment may be an active space inside an Organization instance (property *isLocatedIn*), on the other hand an Organization instance may be located in a large size instance of an Environment (property *isIn*). This Environment intances describe not only physical spaces, but also some kind of functional spaces that are common in different organizations (for instance, a classroom is the same kind of space both at PUC-Rio, in Brazil, or at LIP6, in France).

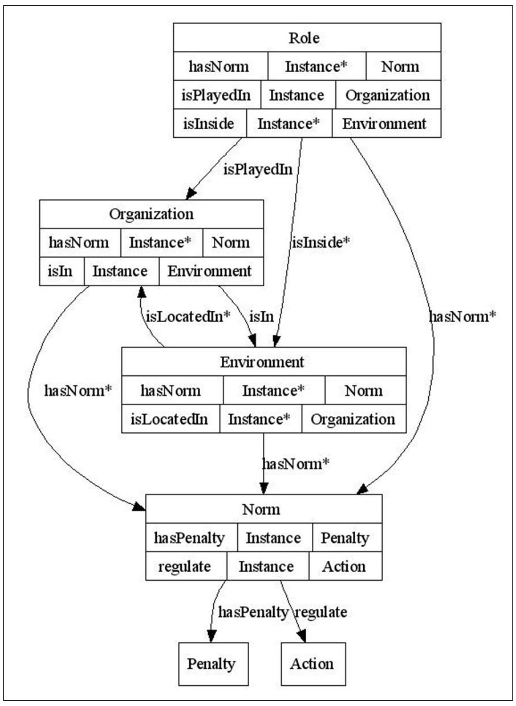

Figure 2. Extended ontology scheme

A Role may be functionally associated with a specific Organization (property *isPlayedIn*). On the other hand, the property "isInside" defines the physical position of a role inside an Environment instance, which is intrinsically related to a symbolic region provided as location information by the MoCA services. While the ontology may have norms associated to each instance of its different classes and subclasses(property *hasNorm/*), the way in which these norms are composed to infer the set of applicable norms is described by a set of rules. A rule inference engine reads the data associated with the instances of the normative ontology and, based on the rules of composition, produces the set of norms applicable to a given situation [Felicíssimo et al. 2006b].

As we show in Figure 1, in our proposed architecture, a Monitor Agent (MA) is responsible for a given space. It subscribes to the MoCA services (through MoCA/MAX) to be informed whenever any User Agent (UA) — associated with a specific Role instance — comes inside (or goes outside) its supervised space. When any person enters (or leaves) that space, MoCA notifies the MA about this event, identifying the associated mobile device. Hence, using DynaCROM, this MA is capable of inferring the set of norms applicable to a given User Agent inside the monitored space, according to what is described in the ontology and in the specific set of rules. This set of norms will regulate the interaction among the UA and the agents responsible for providing the services available in that space (A_1 to A_n).

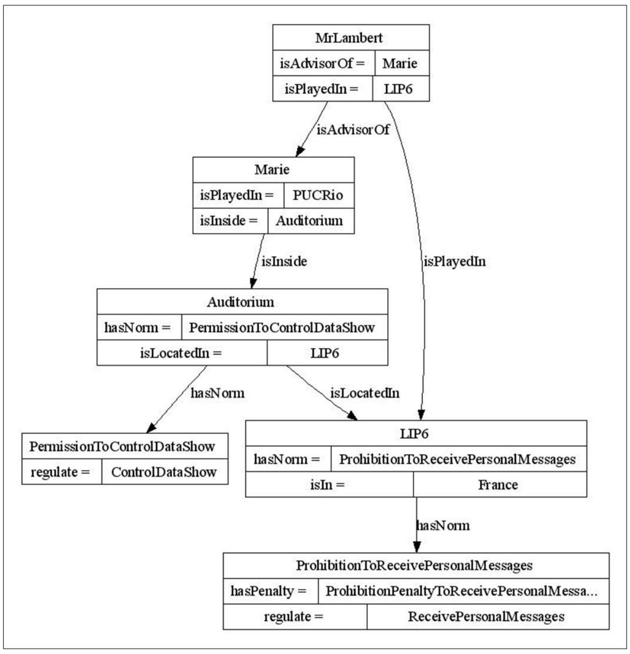

Figure 3. Relationship between instances of our case study in a specific scenario

6. Case Study

The main idea of this case study is to provide specific situations to show the inference of the applicable norms based in complex rules, using the dynamic context information provided by the MoCA services. In the scenario described here, we consider several universities where we have some common Environments such as classrooms, professors' rooms, students' rooms, and so on. The Roles are played by professors and students who may go across different environments in the same university or in different universities. We specified all the presented norms by instantiating our normative meta-ontology. The ontology instance extended the meta-ontology with new concepts related to the representation of the interaction norms and roles. We then populated the ontology with instances of the existing classes and subclasses. For this case study, our University instances are PUC-Rio, in Brazil, and LIP6, in France. While in PUC-Rio we have Mr. Silva playing the role of a professor and Gabi as student, in LIP6 we have Mr. Lambert as a professor and Marie as student.

Marie, who is a computer science student at LIP6, is travelling to Brazil to spend one year at PUC-Rio as a visiting student. Before coming to Brazil though, she is giving a lecture in which she presents her study plans for the period. Figure 3 represents this situation, showing the relationships that are valid between instances for the specific scenario we described. As we see in the figure, Marie *isInside* the Auditorium (instance of Environment), which *isLocatedIn* LIP6 (instance of Organization), and both instances have norms associated specifically to each of them. In our approach, the set of norms applicable to Marie will be composed dynamically, based on rules defined for each environment. We could assume, for example, that being the Auditorium a public space, besides the norms specific for that space all norms defined for LIP6 should be applied. Then Marie would be allowed "to control the datashow", but not allowed "to receive personal messages" while inside the Auditorium.

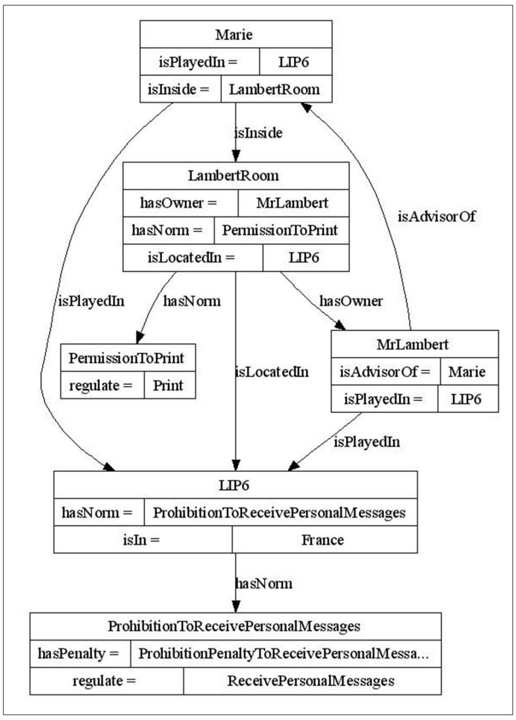

Figure 4. Another example showing instances in a second situation

Supposing that after the lecture, Marie goes up to the room of professor Lambert, her advisor, then we have a different scenario, because the property *isInside* of Marie changes to describe the new situation. Now we have Marie inside the room of prof. Lambert (instance of Environment), which *isLocatedIn* LIP6 (instance of Organization). As shown in Figure 4, other relationships are valid in this situation. Lambert *isAdvisorOf* Marie and *isTheOwner* of that specific room. Thinking of that environment as a private space, we could suppose that the rules that describe the composition of norms in such environment don't include the norms specific for LIP6, and as Lambert is the owner of the room and also the advisor of Marie, all norms that are related to the room would also apply to Marie. Then in this case she would be allowed "to use the printer", to print a report.

In a final example, we consider that Marie went to PUC-Rio, as a visiting student, and is attending a meeting at the room of professor Silva, together with Gabi, a Brazilian student. As shown in Figure 4, in this new scenario, the property *isInside* of Marie, Gabi and Mr. Silva is associated with Mr. Silva's room. In addition, we have that Mr. Silva *isAdvisorOf* Gabi, both Gabi and Mr. Silva roles "are played in" PUC-Rio, and Marie *isVisiting* PUC-Rio. For this situation we assume a hypothetical set of rules that state that in a private room the organizational norms apply only for those that are connected to that Organization instance by the property *isPlayedIn*, but the norms of the Environment instance apply to all. As a result, all of them are prohibited "to receive personal messages", but only Gabi and Mr. Silva are allowed "to use the printer".

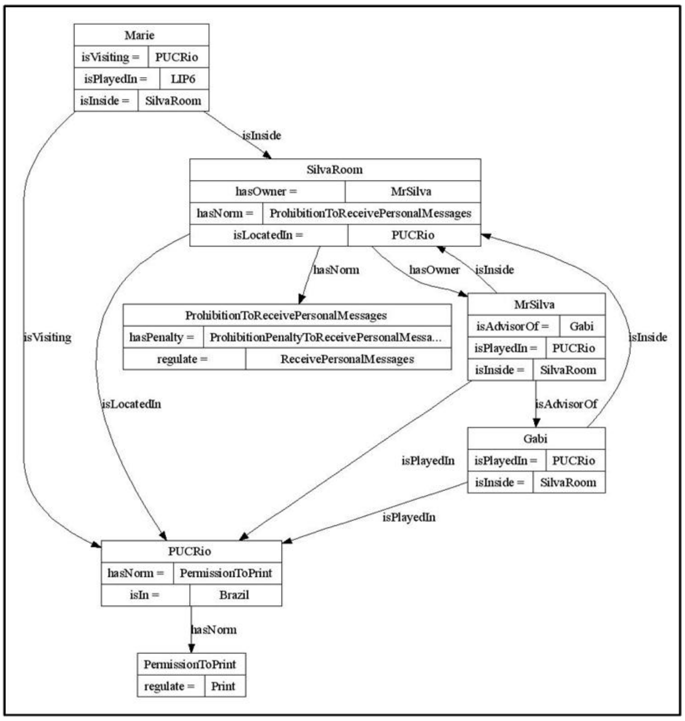

Figure 5. Another example showing instances in a third different situation

7. Implementation Status

Our case study was implemented using Java, the JADE framework, the Jena API and the MoCA/MAX API. Our agents were implemented using the JADE framework. The Jena API was used as a programmatic environment for OWL and as a rule based inference engine, with the rules written following the Jena rule syntax. The MoCA/MAX API was used to implement the subscriptions to the MoCA services. To extend and instantiate the normative meta-ontology the Protégé Editor was used.

More precisally, a normative behavior was implemented for the Monitor Agent using Jena's necessary classes and methods. This behavior makes the agent capable of interpreting the ontology and the rules, both described in files available at a known location. The MA subscribes to MoCA/MAX to be notified whenever a mobile device (running a User Agent associated with a Role instance) enters or leaves the supervised space. Everytime the MA is notified about a UA entering the space, it composes the set of applicable norms executing the normative behaviour.

8. Conclusion

This work proposes a system that applies regulatory mechanisms to coordinate the interaction among heterogeneous entities in an ubiquitous computing scenario. As a norm model, we used the DynaCROM approach, which provides a regulatory mechanism that supports the dynamic composition of norms based on rules and on context information. This mechanism is flexible, easy to operate, and permits that norms are created, deleted and modified at runtime. For managing the context information necessary to implement this system, we used MoCA architecture and MoCA/MAX extension.

In our norm model we consider four general scopes of norms: Environment, Organization, Role and Interaction. Instances of these regulatory scopes have norms associated with them, and rules of composition describe how these norms are dynamically composed to produce a final set of norms. In our approach, while the Environment scope has a topological semantics, Organization, Role and Interaction scopes account for the social aspects. While current ubiquitous support is mainly concerned with topological aspects, we provided a way of considering the social context and its influence in the entities' interaction process. In systems like Oxygen and Aura, for instance, it is only possible to adapt ubiquitous services to the location and the preferences of the users. On the other hand, using our approach we are able to define complex rules that will determine norms of adaptation depending not only on the topological context (location, for instance), but also on all the instances involved in a given situation and the relationship among them. This is what we highlight as the main contribution of our work.

A small scenario was introduced and implemented as a first example for illustrating and testing our approach. We are currently expanding our scenario, and will start conducting real experiments with physical mobility. We also plan to refine the integration of our architecture with an agent architecture, such as BOID [Broersen et al. 2001] or JADEX [Pokahr et al. 2003], and also alternatively with an external mediator architecture, such as XMLaw [Paes et al. 2004] or LGI [Minsky and Ungureanu 2000].

It is important to mention that some conflict may arise when composing a final set of norms using norms belonging to different scopes. However, we didn't deal with this matter in this paper, letting the subject to be analysed in further studies.

References

Bakan, J. (2003). The Corporation, film directed by Mark Achbar, Jennifer Abbott and Joel Bakan. Big Picture Media Corporation.

Bellifemine, F., Poggi, A., and Rimassa, G. (2001). JADE: a FIPA2000 compliant agent development environment. In *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*, pages 216–217.

Broersen, J., Dastani, M., Hulstijn, J., Huang, Z., and van der Torre, L. (2001). The BOID architecture: conflicts between beliefs, obligations, intentions and desires. In *Proceedings of the International Conference on Autonomous Agents*.

Felicíssimo, C., Choren, R., Briot, J.-P., and Lucena, C. (2006a). Supporting regulatory dynamics in open MAS. In *Proceedings of AAMAS06 - Workshop on Coordination, Organization, Institutions and Norms in Agent Systems (COIN-06)*.

Felicíssimo, C. and Lucena, C. (2005). An approach to regulate open multi-agent systems based on a generic normative ontology. In *Proceedings of the 1st Workshop on Software Engineering for Agent-oriented Systems (SEAS 2005)*.

Felicíssimo, C., Lucena, C., Briot, J.-P., and Choren, R. (2006b). An aproach for contextual regulations in open MAS. In *Proceedings of AAMAS06 - Workshop on Agent-Oriented Information Systems (AOIS-06)*.

Harroud, H., Khedr, M., and Karmouch, A. (2004). Building policy-based context aware applications for mobile environments. In *MATA*, pages 48–61.

Johanson, B., Fox, A., and Winograd, T. (2002). The interactive workspaces project: Experiences with ubiquitous computing rooms. *Pervasive Computing Magazine*.

Kindberg, T. and Fox, A. (2002). System software for ubiquitous computing. *Pervasive Computing Magazine*.

Minsky, N. H. and Ungureanu, V. (2000). Law-governed interaction: a coordination and control mechanism for heterogeneous distributed systems. *ACM Transaction on Software Engineering Methodology*, 9(3).

Nascimento, F. N., Sacramento, V., Baptista, G., Rubinsztejn, H. K., and Endler, M. (2006). Development and evaluation of a positining service based in ieee 802.11 (in Portuguese). In *Proceedings of the XXIV Brazilian Symposium on Computer Networks (SBRC 2006)*.

Paes, R., Carvalho, G., Almeida, H., Lucena, C., and Alencar, P. (2004). A conceptual architecture for law-governed open multi-agent systems. In *Proceedings of Argentine Symposium on Software Engineering (ASSE2004)*.

Pokahr, A., Braubach, L., and Lamersdorf, W. (2003). Jadex: Implementing a BDI-infrastructure for JADE agents. *EXP - In Search of Innovation (Special Issue on JADE)*, 3(3):76–85.

Rubinsztejn, H. K., Endler, M., Sacramento, V., Gonçalves, K., and Nascimento, F. N. (2004). Support for context-aware collaboration. *First International Workshop on Mobility Aware Technologies and Applications (MATA 2004)*, 5(10):34–47.

Rudolph, L. (2001). Project Oxygen: Pervasive, human-centric computing - an initial experience. In *Proceedings of 13th International Conference in Advanced Information Systems Engineering (CAiSE 2001)*.

Sacramento, V., Endler, M., Rubinsztejn, H. K., Lima, L. S., Gonçalves, K., Nascimento, F. N., and Bueno, G. A. (2004). MoCA: A middleware for developing collaborative applications for mobile users. *IEEE Distributed Systems Online*, 5(10).

Saha, D. and Mukherjee, A. (2003). Pervasive computing: A paradigm for the 21st century. *Computer Magazine*.

Soldatos, J., Pandis, I., Stamatis, K., Polymenakos, L., and Crowley, J. L. (2006). Agent based middleware infrastructure for autonomous context-aware ubiquitous computing services. *Journal of Computer Communications*.

Sousa, J. P. and Garlan, D. (2002). Aura: An architectural framework for user mobility in ubiquitous computing environments. In *Proceedings of 3rd IEEE/IFIP Conference on Software Architecture*.

Weiser, M. (1991). The computer for the twenty-first century. *Scientific American*, 265(3):94–104.