Role-Based Collaboration Extended to Pervasive Computing

Kahina Hamadache, Luigi Lancieri

Orange Labs, 42 rue des Coutures, 14000 Caen, France
LIFL, University Lille 1, Lille, France
kahina.hamadache@orange-ftgroup.com
luigi.lancieri@univ-lille1.fr

Abstract

Computer Supported Collaborative Work (CSCW) has been a hot point in researches for some decades. Recent progresses in software and hardware technologies have allowed the use of more and more advanced applications and services. It also brought computing capabilities to mobile devices such as smartphones and laptops. This has led to an extensive use of computers to collaborate in some unexpected manners. Among the abundance of models designed to support collaboration some are particularly promising: tasks models, roles models and collaboration's context models. Simultaneously the Pervasive Computing paradigm has emerged from recent researches. In this paper we propose a model to integrate the pervasive computing perspective into the collaborative work. This integration is proposed by the use of an original model: the PCSCW model (Pervasive Computing Supported Collaborative Work). This model relies on some robust concepts: a role model inspired by some recent works, a classical task model coupled to a precise resource model and the development of device collaboration rules. The resulting model provides a seamless and transparent cooperation of devices to simplify and facilitate the collaboration of humans.

Key words: Pervasive Computing, Collaborative Work Modelling, Role-Based Collaboration Model

1. Introduction

In the past few years, computer supported collaborative work (CSCW) has became an unavoidable aspect of everyday's life of most companies. Actually, computers can channel the collaboration between people in many ways; they facilitate many tasks and allow monitoring and regulating of the collaboration. Most simple tools for collaboration already exists, even if those are still young and lack of maturity they offer the basis for simple collaboration. However, to provide more advanced collaboration features and opportunities, it is necessary to go deeper in this research field.

Research in CSCW is an “old” domain (relatively to computer sciences history) and many researchers have focused on the design of collaboration model with the final objective to allow a better management and understanding of collaboration. Thus many interesting models have been carried out in the history of collaborative software, but as for now none of them have ever really be able to overcome others and to solve all issues. However we think that some models are more relevant and promising. This is the case for role-based models. These models focus on the simple but still unavoidable concept of roles in collaboration. A role can be defined in terms of responsibilities and rights where responsibilities are actions a role player must perform and rights are actions he can perform. Zhu and Seguin [13] characterise a role in collaboration as follows:

- A role is independent of persons;
- A role should consider both responsibilities when the human player is taken as a server and rights when the human player is taken as a client.
- A role has to accomplish the tasks specified by the responsibilities.
- A role can be performed by one or many human players at the same time. A role can be created, changed and deleted by a human user with a special role.

Another evolution we can observe nowadays is that electronic devices are becoming more and more common, compact, smart and autonomous. Thus, the digital environment is taking each day a greater part in our world, leading us to a digital augmented environment. In this vision, the pervasive computing paradigm is a major domain, it aims at making all these smart electronic devices spread in our environment collaborate to provide us a seamless interaction with the digital world.

A natural extension for CSCW is then to be more suited for this kind of environment where the number and the type of devices is nothing more than a variable to which you have to be able to adapt. In this perspective, an interesting challenge for CSCW is to know dynamically and efficiently how to take advantages of this technology to improve the collaboration between humans. Potential advantages are various, we may think to simple mechanisms such as smart device control via a computer until complex mechanisms of collaborative context awareness.

However, only few advances have been made toward the better integration of pervasive features in CSCW systems. Thus the main objective of our work is to provide a model to natively support the pervasive computing paradigm inside the collaboration of users. As we will see in the next sections this model relies on several sub models: a role model, a task model and a resource model. Those models are

coupled with the definition of "device collaboration rules" that define automatic device behaviours according to a given context.
In the remainder of this paper, we present our role-based model for pervasive computing supported collaborative work. We start with a presentation of already existing collaboration models for CSCW based on tasks, roles and collaborative awareness. Once all this notions have been introduced we are able to present our Pervasive Computing Supported Collaborative Work model, its main principles and two illustrative use cases. Following our researches chronology we try to anchor or model in a more concrete perspective by presenting its dedicated simulator, this last point allows us to introduce basics about CSCW evaluation and the way we consider it to validate and evaluate our model.

2. COLLABORATIONS MODELS FOR CSCW

The main purpose of CSCW systems is to handle the collaboration between users. To do it they can rely on technical evolutions and tools adaptation for multiple users. But collaboration raises problems that are going far beyond technical issues. Indeed, the main problem of CSCW is the collaboration itself, how a system can effectively support collaboration patterns and how it can be aware of the current collaboration status. To tackle these problems several models of collaboration have been proposed. Among them, some kinds sounded more promising: tasks models, roles models and the refined collaborative awareness models. In the following we will illustrate these models by presenting some of their related researches.

2.1 Task models

The task model is now widely accepted by the CSCW community to be one of the bases to represent collaborative work. It has been the subject of many articles and is still an active research field. Task models' goal is to identify useful abstractions highlighting the main aspects that should be considered when designing interactive systems. The main advantage of task models is that they represent the logical activities that an application must support in order to better understand the user, his work and his expectations.
Keeping in mind the necessity to improve the general ergonomics of collaboration, Molina et al [4] proposed a generic methodology for the development of groupware user interfaces. This approach is called CIAM (Collaborative Interactive Applications Methodology); the approach defines several interrelated models capturing roles, tasks, interactions, domain objects and presentation elements. Even though, the models cover all important aspects involved in groupware user inter-

faces, they are only used at the analysis stage. Subsequent development phases (e.g. requirements or design) are not covered. The methodology is not assisted by a tool which would facilitate the creation and simulation of the various models. In particular, the latter is an important shortcoming since the animation of models is an important technique to obtain stakeholder's feedback. These works fail to account for user roles and multiple role-based views on the same collaborative task. Aiming at fulfilling this gap Vellis [11] has adopted an extended version of CCTT (Collaborative ConcurTaskTrees [16]), which would be taking care of user role differentiations and their effect in the whole process.
In a different perspective Penichet et al [9] propose a task model for CSCW based on the use of several well known task modeling aspects. Their model is aimed at describing "the tasks that should be performed to achieve the application goals" by giving them a good characterization. This model is aimed at designers that have to design groupware systems. What they propose is not a complete new model of tasks but a new "composition" of existing tasks models in order to have a better, more complete and more effective task model. Their approach is based on the description of tasks that are realized in groupware systems keeping in mind more classical aspects and mechanisms to analyze them. They argue that classical CSCW features or time-space features are not enough to correctly describe a groupware, but that a well done combination of them can do it.
Task models are interesting, because they can be easily understood by humans as they represent "classical" organization of collaborative work. But some models take different ways to represent the collaboration, making them interesting by the simple fact that they have new points of view of the collaboration.

2.2 Collaborative awareness models

Collaborative awareness is the capacity to be aware of the current state of the collaboration. It can be useful for humans when they need to coordinate themselves to accomplish a specific task or schedule an operation. But we know it can also be useful for devices in order to behave properly. Thus collaboration awareness is critical for most of collaborative systems as it helps maintaining the coherence of the collaboration and eases the work of collaborators. Let's have a look at some researches of this domain.
Drudy and Williams [17] proposed a cooperative awareness model based on role, but the relation in roles' cooperation was not mentioned in their paper. Gutwin and Greenberg [18], proposed a workspace awareness framework, this framework describes three aspects: its component elements, the mechanisms used to maintain it, and its uses in collaboration. These parts correspond to three tasks that the groupware designer must undertake in supporting workspace awareness; understand what information to provide, determine how the knowledge will be gathered, and determine when and where the knowledge will be used.

In an analytic approach Yan and Zeng [10], [20] proposed an original model for group awareness inside CSCW systems. They assume that there are mainly two aspects in group awareness: "group awareness model" and the "method of realization". They point out the fact that for now, the main problem is still the construction of a robust model.
What they want to do to solve the resisting problems is to analyze basic elements of group work: "task", "action" and "role".
The application they developed is composed of a set of modules, each one dedicated to a specific goal, but the more interesting of them is obviously the task disassembling one. The elementary definition here is the formal definition of a task as a triplet *T: (Role1, Action, Role2)* where T is the task, Role1 and Role2 are roles associated with the task. Role2 is only mandatory when Role1 cannot complete independently the Action. So, to disassemble a task, the system recursively disassemble Role1, Action and Role2 until it can't divide any more. When it reaches this state the task is defined as "atomic". They also define a set of rules for disassembling in order to avoid inconsistent state. Moreover they notice that task disassembling is time consuming and then propose to pre-process most common tasks categories into task tree templates.
Thus they provide templates to represent tasks and then user's activity, allowing them to have an interesting description of current collaboration.
Moreover they reasoned by telling that as task depends on role, task is a more efficient group awareness descriptor. But they also showed that task depends on action. If we follow their argumentation we should say that action is an even more effective descriptor, than they do not mention. Finally, we do not really agree to the task representation they propose. By example we can say that, depending on the group in charge, a task will not be handled the same way, with the same roles and actions, even if the goal is the same. That's why we think that task and roles should not be depicted by some stilling trees and templates but that it should be dynamically extracted and modified during the group collaboration.
While this precedent work relies on the precise dissection of tasks and roles, Rodden [12] proposes a model of awareness for cooperative applications measuring the awareness intensity by the flow of information between application programs. If this model successes in describing group awareness characteristics, it does not really include group structure into its measure.
Researches often want to be as generic as possible, in order to produce a model able of representing any kind of collaboration. Thought, generic implies less coupling with the domain, then most of the times it is necessary for models to focus on a specific domain.
We are reaching the roles models, which are the ones that motivated use for this research. Roles can seem simple, but describing them correctly with all their characteristics is a really complex issue.

2.3 Role-based collaboration

Role-based collaboration (RBC) is a methodology to design and implement new computer-based tools. It is an approach that can be used to integrate the theory of roles into CSCW systems and other computer-based systems. It consists in a set of concepts, principles, mechanisms and methods [1]. RBC is intended to provide some benefits to long-term collaboration: identifying the human user “self”, avoiding interruption and conflicts, enforcing independency by hiding people under roles, encouraging people to contribute more and removing ambiguities to overcome expectation conflicts. It is also intended to provide benefits to short-term collaboration: working with personalized user interfaces, concentrating on a job and decreasing possibilities of conflicts of shared resources, improving people’s satisfaction with more peoples’ playing the same role during a period and transferring roles with requirement of a group. Finally, in management and administration, it helps at decreasing the knowledge space of searching, creating dynamics for components and regulating ways of collaboration among parents.

Some CSCW systems have indeed applied the concept of roles. Barbuceanu et al [6] have proposed role based approaches to agent coordination. This approach includes a "practical, implemented coordination language for multi-agent system development" that defines, agents, their organization and roles. Agents play roles in an organization, and a role is defined by its major function, permissions, obligations, and interdictions. A role’s permissions include agents under its authority and its acquaintances. An agent’s beliefs and reasoning are partitioned on the basis of the roles it plays to facilitate context switching [6]. A combination of events leads to a situation for the organization, with each agent member in a given local state. An agent’s behavior in a situation is determined by its conversation plans, and these are usually specified to be between a particular pair of roles.

Edwards [7] propose a system that can implement a variety of useful policies in collaborative settings, particularly in the areas of awareness and coordination. This system uses the notion of roles to associate categories of users with particular policies. Intermezzo roles can represent not only groups of users, but also descriptions of users in the form of predicates evaluated at runtime to determine group membership. Dynamic roles, in particular, expand on one of the central themes in this work: by bringing information about users and their environments into the system, it can make computer augmented collaboration more responsive, and can free users of many of the implicit burdens in working with today’s collaborative systems.

In a more recent article, Zhu [2] proposes his view of collaborative authoring based on the use of roles. He points out the fact that collaborative systems should not only support virtual face-to-face collaboration between distant people, but should also improve physical face-to-face by providing mechanisms to overcome drawbacks of face-to-face collaboration. They notice that WYSINWIS (What You See Is Not What I See) can be an efficient model for the development of collabor-

ative systems. Thus WYSINWIS systems can allow different users to have different views of a shared document according to their roles inside the collaboration. This kind of interaction is not totally new, and such systems exist for a long time, but what they propose is a mechanism based on the precise role definition and specification to allow roles to be dynamically tuned and managed in the system.
Furthermore, Zhu and Tang [3] propose a role based hierarchical group awareness model (RHGAM). Firstly RHGAM constructs a group cooperation environment (GCE), and then GCE is extended by group awareness content, awareness hierarchy, the task decomposition rule. The model divides the awareness information into four levels by decomposition and recombination using a role-task graph and the thinking of group structure. In RHGAM, role is the basic of group cooperation; with the different group structure and task relation, the awareness information is shared between roles hierarchically.
In a relatively different perspective, Ahn et al [5] implemented a role-based delegation framework to manage information sharing (FRDIS) for collaborating organizations. Their central idea is to use delegations as a means to propagate access to protected resources by trusted users.
Role models propose a "natural" approach to collaboration; with the help of task models it is possible to have an accurate description of user's collaborative work. Still we want to go further and properly consider and integrate devices as part of the collaboration. To do it we propose our model based on the description of roles, tasks, actions, resources to perform them and available devices' resources.

3. PCSCW collaboration model

Keeping in mind the works that have been done in the different domains we're interested in, we propose our own model relying on some simple concepts: tasks, actions, roles and resources. The main principle of this model is the following: we rely on the fine description of roles, tasks, actions, resources required and the available devices' resources; then by a simple comparison of required and available resources we can select the right "device collaboration rule" to make devices collaborate seamlessly and facilitate the collaboration of users. We'll now give further details about these main aspects and their use in the process of making devices automatically and smartly collaborate.

3.1 Task

The first concept to define is the task. This concept is one of the most popular of the recent researches in collaboration modelling. A task can be defined as a set of actions to be performed in a specified or unspecified order to fulfil the task objective. In addition, a task is not always (and in fact most of the time is not) an atomic one, meaning that it can be composed of several sub-tasks with their own actions and objectives. Moreover, we can point out that the collaboration of people takes place when they need to perform a task they can't or shall not do alone. If this task has to be performed by more than one person, it can be considered has a "shared task" or a "common task".

3.2 Actions

Actions can be seen as tasks components. In some perspective they could be considered as atomic tasks, however we think that a task carries its own meaning, actions don't, and that's why we should consider them as sub-atomic tasks. To illustrate this idea we can figure that the action "opening a web browser" has no "meaning", but opening a web browser and writing a word in a search engine has its own meaning, it is the task of "searching on the web".

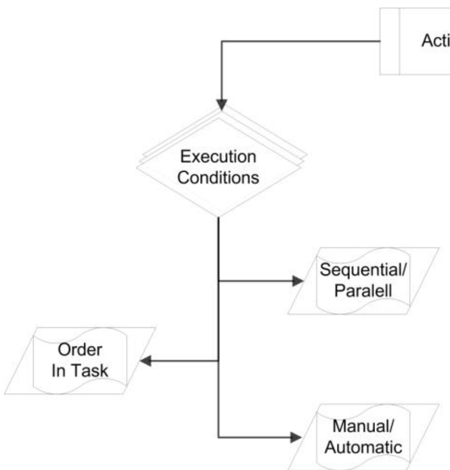

Fig 1. Action Specification.

3.3 Role

As we have seen previously, a role can be defined as a set of tasks to be performed by a single entity, giving it responsibilities, rights and duties. A role is not reserved to persons; it can also be played by a group of persons or by an entire organization. Besides, in the same way as a role can be designed for more than one person, a person can play several roles at a time. This is particularly true in the case of a person belonging to multiple groups (for example a work team and a sport club). In addition, a role can have a specific "cardinality" inside of a group, meaning that you can have several people in the same group playing the "same" role. This aspect of the role concept can be confusing if you consider that two people never do the exact same work, that's why roles have not to be confused with peoples.

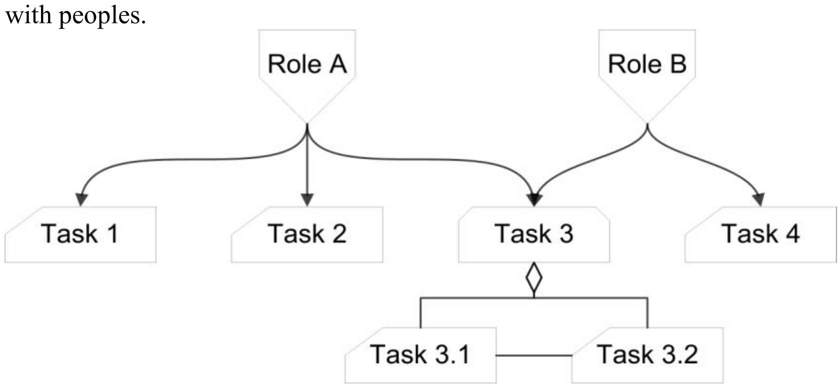

Fig 2. Roles and Tasks.

On figure 2 we give an example of how roles can interact through their allocated tasks. "Role A" has 3 tasks: 2 are dedicated to it and the third is shared with "Role B". Then this task two is subdivided in 2 subtasks: "Task 3.1" for Role A and "Task 3.1" for Role B. One could have proposed to remove Task 3 and just leave tasks 3.1 and 3.2 affected to their roles. Still we argue that for some of them it is necessary to preserve links between related tasks. Indeed some tasks require several roles to be completed. For instance the task "writing software specification" is composed of two subtasks: "writing software business specifications" and "writing software technical specifications". Besides, you can't write technical specifications before business ones have been written and they can't be always written by the same person, then the two roles associated to this task will not be held by the same person.

3.4 Resources

If you intend to model the context of people in order to develop context awareness mechanisms, at some point of your reflection you will have to face the representation of users' resources relevant for the part of context you're interested in. Obviously in our model, we can't avoid this part, it is in fact one of the most interesting point we want to explore. Indeed, we argue that the description of tasks should be made through the representation of resources required to perform it. Going even further we could describe facultative resources that can be effective to perform the task but which are not mandatory. Thus, considering that you've got a fine description of the task a group is performing, you can have accurate indicators of the state of the task. This could lead to a fine monitoring of the task and then to a fine collaboration awareness mechanism.

3.5 Smart devices

By extension of the precedent aspect of our model, we propose to associate tasks to (smart) devices. To do it we have to figure out that smart devices are parts of the available resources. Furthermore, it is necessary to have a description of devices capabilities. For instance, if you consider that a high-speed connection to the Internet is required for your task, the best device to support it can be quickly identified by a simple query to available ones or by a more efficient request to some kind of a context manager. Such a mechanism is particularly effective, as it can make several devices cooperate seamlessly.

3.6 Roles for devices

As a refinement of devices description, we propose to define their roles in the collaboration process and more precisely for a task. Thus, a task becomes the natural link between peoples and devices via the description of roles for both of them. As for "Human Role", the role of a device describes its responsibilities and rights. To take a simple example a device can have a role in a collaboration process giving the responsibility of providing the Internet connection for a given user. With this example we can point out a major difference between human roles and devices roles, humans' roles are based on actions performing while devices roles are based on resources providing. The Fig 3 sums it up. On this figure we quickly modelled a simple, but common task: the development of a client-server application which implies the development of a shared object: the communication interface between the client and the server. For this task we need a Collaborative Design Tool and an Instant Messaging Tool, which is not mandatory but can improve the collaboration. An interesting point here is that the model can enhance the collaboration by proposing optional resources such as, in this case, a messaging tool. Furthermore the model itself can be refined by describing precise rules for the messaging tool to be proposed and used; in some cases it can be preferable not to use it. The designing tool is provided by a computer while the messaging one is available on the smartphone. Thus we can say that the computer plays the role of "Heavyweight application provider" while the smartphone has a "Messaging application provider" role.

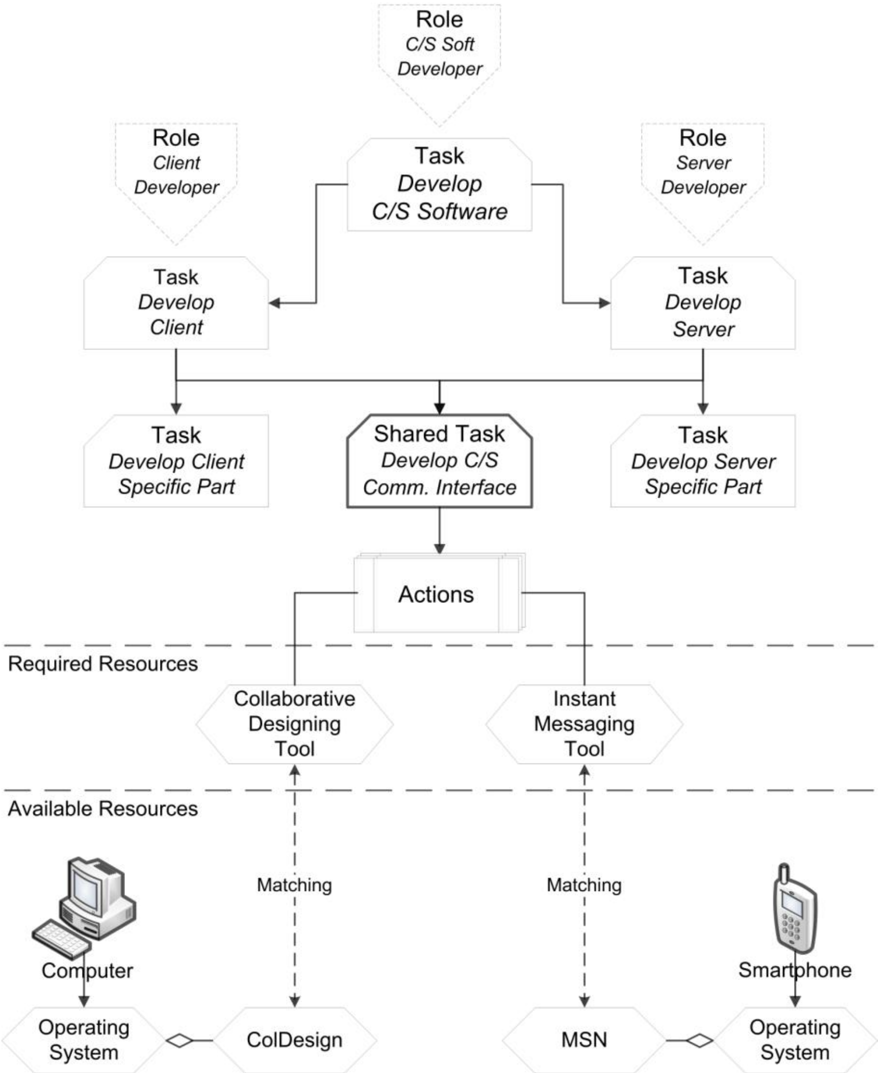

Fig 3. PCSCW model, example of application.

3.7 Devices collaboration rules

All devices don't natively support collaboration with others. In order to solve this kind of issue we argue that the definition of device collaboration rules could be of great help. These rules intend to define tasks that could be automatically performed by devices to collaborate in order to allow a user to do its own task. The main idea behind this is the following: a user needs two (or more) resources to complete an action related to a task; these resources are not available on a single device, but the combination of several of them can supply the resources. Thus, device collaboration rules define what actions can be performed by devices to collaborate, finally providing required resources to the user. These collaborations can be of various kinds: network access sharing, heavy computing task delegation and notification of events, anything you can imagine to make several devices cooperate. In a previous paper [19] we defined that each device collaboration rule is written with the following syntax (1):

$$\text{IF } (\textit{context.resources} \equiv \textit{rule.resources}) \text{ THEN DO } \textit{rule.behavior} \quad (1)$$

This simple syntax summarizes how device collaboration rules are working. The first part of this formula corresponds to the comparison between the current state of the context and resources required to trigger the device collaboration. The second part of the formula corresponds to the collaborative behaviour of devices. Thus if context resources are matching rule resources we trigger the related collaboration of devices. Even if this basic mechanism is a critical part of our model we've established in [19] that it is not sufficient to completely manage. Indeed if we consider the fact that several rules can have the same set of required resources to be triggered or that a single rule can trigger several behaviours at once, we need another mechanism to decide which option has to be used. In order to solve this issue, we propose to define constraints on resources as triplet {P, V, C} where: *P* a parameter which represents the precise point to be evaluated, *V* the expected (or required) value (or threshold) for this parameter and *C* the criticality of this parameter which represents the relative importance of this constraint. In order to facilitate the understanding and the use of criticality we have defined standard levels of criticality: *Optional*, *Very Low*, *Low*, *Average*, *High*, *Very High* and *Mandatory.* It allows us to quantify, estimate, compare and then choose between several candidate rules. In addition to this triplet we propose to organize constraints in five main categories, facilitating and guiding rules designer in their work: *Availability, Cost, Privacy, Reliability and Security*. These constraints describe desired characteristics of resources according to the tasks to be performed. Given these constraints we've got a mean to know which behavior is the most suited in the current state of the context. Thus to decide the most adapted we only have to evaluate the suitability of the potential rules according to their respect of the defined constraints. We will illustrate this mechanism in the second example of the following section.

4. PCSCW in action, two use cases

To illustrate our model, we can consider a use case that we already mentioned in [8]. This example relies on the Pervasive Brainstorming systems we developed in this previous article and can be described as following:

- The manager of a team wants to have the opinion of its team members about a specific topic (for example about a project he's planning);
- As his team is often spread over different locations, he can't meet each of them physically;
- To solve issues they can encounter for this collaboration, we proposed a service based on the use of mails and forum to channel the opinion of the team and provide efficient synthesis of the group opinion;
- The system itself is based on the automatic publication of multiple-choice questionnaires which can be sent by mail to a dedicated mailbox, mails are then analyzed and contained questionnaires are published on a forum where team members can vote and give their opinion.

If we consider this use case with our model we can distinguish two roles: the manager role and the basic team member role. The manager role has a cardinality of 1 while the team member role has an unspecified cardinality for this group. The team itself is mapped to a "Group Role" with its own set of tasks. The manager role allows its player to perform a "Send New Questionnaire" task while team members are allowed to perform the task "Answer a Questionnaire".
Let's consider the case where a member of the team, Bob, is equipped with a laptop and a cell-phone, both switched on. His laptop has only a Wifi and an Ethernet adapter without available network in range. On the contrary, his cell-phone is connected to a HSDPA network and has its own Wifi adapter (but as for the laptop, without access point available). Bob's manager has just sent a new questionnaire; an automatic mail is sent to him with a link to the published questionnaire. In the traditional case, Bob should open the mail, go on the forum and access to the questionnaire on his phone. We suggest that this interaction could be dramatically improved. Let's consider that Bob is deeply focused on his laptop and that his cell-phone lies at some distance of him. Here a simple but still efficient device collaboration rule can take place:

- When receiving a new mail on cell-phone;
- If User is working on superior ergonomics device (Computer, Laptop …) which can be connected with cell-phone;
- Then perform tasks: establish a bridge connection from cell-phone to computer and notify user of the new mail.

Fig 4 represents what resources are necessary to perform the "Bridge Connection" task.

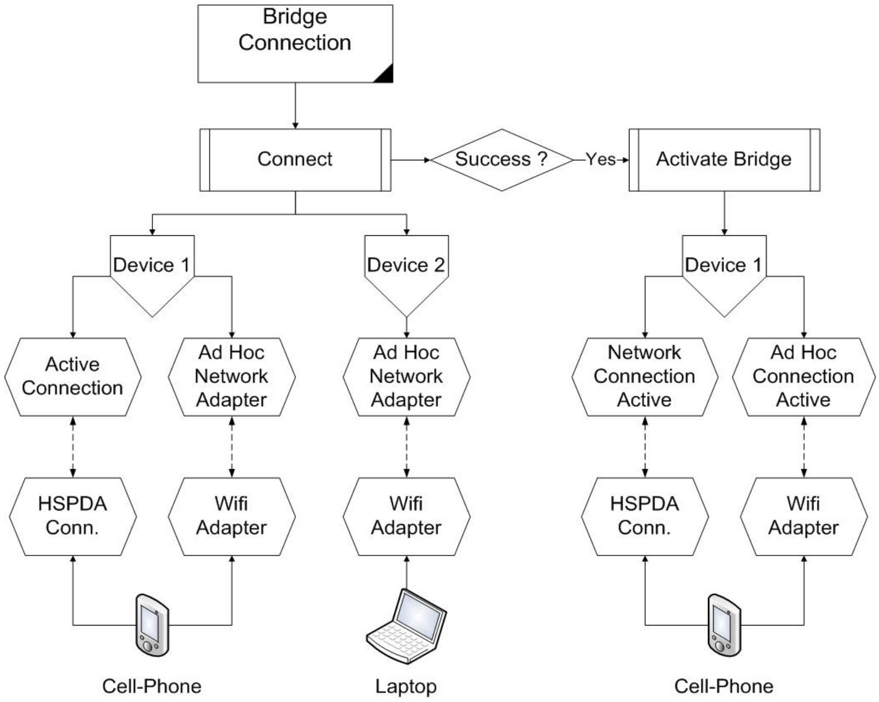

Fig 4. Device collaboration, connection bridging.

As we can see on the previous figure our model serves at representing resources of users' task to find matching devices but also to determine if those devices can cooperate to supply the desired resources. Indeed to perform the connection bridge between two devices we have defined the required resources: an active connection on the device that have to bridge it; and a network adapter of the same type on both of devices. In our case the available connection can be found on the cell-phone as the HSPDA one while common network is supported by Wifi adapters. Once the cell-phone and laptop are connected the bridge can be activated.

In order to illustrate more precisely the device collaboration rules and their relative mechanisms we can consider a second example. This second use case implies the collaboration of 3 coworkers. Leela, Amy and Philip are members of a team and have to collaborate on a new marketing campaign for the new product of their company. In this perspective they have to perform several tasks together. Let's suppose that they have to make a brainstorming session to design a new advertising board. Amy is working at their main office, but Leela and Philip are not physically present. Leela is working at her home while Philip is in mission in Kenya. In order to be able to work at the same time Amy has sent invitations to Leela and Philip for a virtual Brainstorming with a dedicated software at 3 PM (GMT). In a "device consideration" Amy is working on her usual workstation, Leela has its personal laptop, Philip on his side has a tablet-pc and a smartphone. At 3 Amy has started the server part of the application and has connected her station. At the same time Leela's laptop and Philip's tablet-pc need to connect to the Internet in order to be able to join the Brainstorming platform. To do it they rely on the PCSCW model that should allow their devices to make the right decision. The task associated with the brainstorming activity described with the PCSCW implies several constraints on the resources used by the devices. For the "Connect to Internet" action we've got constraints on several resources. Firstly we've got four constraints on the network connection:

- 1 security constraint: the encryption has to be at least RSA; this constraint has a Very High criticality as the collaboration taking place is close to confidential;
- 1 availability constraint: the average provided bandwidth has to be at least 0,5mbps, this constraint has a High criticality as the application can work with less bandwidth but user's satisfaction and experience may be dramatically lowered by such limitation;
- 1 reliability constraint: the probability to experience network disconnections has to be less than 1 per hour. As this point doesn't completely stop the collaboration it has an Average criticality;
- 1 cost constraint: the price of the connection has to be less than four dollar a minute. As it doesn't obstruct the collaboration this constraint has an Average criticality.

As for the network connection we also have a constraint on the power supply resource:

- 1 availability constraint: the energy supplied has to be sufficient to maintain the connection for three hours in order to have enough time for the brainstorming session. This constraint has a High criticality.

Leela's laptop hasn't many choices and connects itself to the wifi access point of Leela's home's ADSL modem. Philip's situation is totally different. In addition to the tablet-pc, the smartphone and the hotel wifi access point, we've got a description of resources required for the connection to Internet. It also depicts the three possible scenarios to establish the Internet connection:

- Direct connection of the tablet-pc through its satellite network adapter;
- Connection of the tablet with hotel's wifi access point;
- Connection of the tablet with Philip's smartphone with a connection bridge between cellphone's wifi and 3G networks to allow the tablet to acces to Internet.

Each one of these possibilities has advantages and drawbacks:

- Direct connection with satellite network:
    - Advantages: highly secured, only rely on tablet's energy, relatively stable;
    - Drawbacks: slow connection (~0,2mbs) and costly, occasional disconnections;

- Connection to hotel's wifi:
  - Advantages: good bandwidth(~2mbps), free, low energy consuming;
  - Drawbacks: poorly secured (WPA), variable bandwidth, disconnections every fifteen minutes;
- Connection with smartphone:
  - Advantages: as we use ad-hoc wifi the security is up to the two devices and can be relatively good, the average bandwidth is fair (~1 mbps) and the connection is relatively stable;
  - Drawbacks: power supply is limited by smartphone's battery life which is limited to 2.8 hours due to the high energy consumption of the 3G and wifi adapters.

We consider that Philip's tablet has already acquired all these information; he must now find the best solution. This is simply realized by analyzing solutions constraints fulfilments. From the precedent listing of advantages and drawbacks and after having put a score to each constraint in respect with expected and offered values and their criticality we have to come to the conclusion that despite the risk of shortening the brainstorming of some minutes, the smartphone connection option is the one offering the best compromise. Thus the system has been able to automatically find three possible scenarios of connection and has chosen the most adapted solution by a simple comparison between required and available resources and by evaluating the fulfilment of some constraints.

These scenarios show how our model enables the efficient cooperation of surrounding devices to enhance the collaboration of users. Besides, this type of scenario can be further extended to multiple users and devices, creating a real "pervasive and collaborative network".

5. Simulator

For now we have presented our model. However this model has not been tested in real conditions yet. In order to validate it we are currently developing a simulator that will help us in this process. Simulation is a very helpful validation device; it helps understanding the involved collaborations as well as triggers invaluable feedbacks from stakeholders about the elicited requirements. One of our first thoughts was to directly implement a real application to use our model. However, when we studied this project we were rapidly confronted to limitations and difficulties. Among the main difficulties of developing such an application is the necessity to handle various kinds of devices and then a variety of operating systems and environments. Furthermore, even if this problem is not unbearable there is at least another one that can't be simply handled: resources required for each test. Indeed, if you want to validate a model you need to perform batteries of tests, in the case of a real application it can be outrageously costly and time consuming.

Moreover, as we're dealing with multiple users and multiple devices interactions, we would have needed many people in different locations with several smart devices. Thus, we quickly found out that our approach was not the good, and that the only way to efficiently validate our model was to develop a simulator. Benefits of such methods are numerous: you don't need to "hire" a bunch of people and send them on the other side of the city in a fast food with free wireless Internet connection, you just have to click on some buttons; you don't need tens of minutes of preparation before each test, you just have to prepare it once and run it as many times as you want; another advantage here is that you can trace every single event of the simulation and analyze it afterwards, which is much more difficult in real situation.
In order to validate the model we have decided to feed our simulator with collaboration scenarios. As we'll see in the rest of this work, scenarios will be composed of an initial state of the world with a given set of agents and by some events that will trigger collaboration rules among devices. As we already said we will trace the execution of the scenario and the different actions made by agents, in order to be able to replay and analyze them. Thus the simulator will be a real laboratory to construct and improve devices collaboration rules.

5.1 Architecture

The architecture of the simulator has been organized in several modules articulated according to presentations, controllers and data. As the presentation part of the simulator isn't really important and can be changed without interfering with the business aspect of the simulation, we will not spend more time on this part. On the opposite, controllers are the heart of the system. They ensure the management of the application in a technical and business perspective. Most important data of the simulation are stored in an ontology and its associated set of semantic rules.
Technically, as we had chosen to use the JADE[1] framework for the multi-agent aspect we naturally decided to develop our simulator in Java. To access, update and manage the ontology we rely on the Protégé[2] Framework developed at the Stanford University that allowed us to write our ontology in OWL[3]. To be completely useful an ontology needs semantic rules; for this part we decided to follow the recommendations of the W3C and use the SWRL[4] language.
In a "business" perspective, the ontology will be used to represent all data of the scenario, that is to say all information about the initial state of the scenario and all events that are scheduled to occur in the simulation (all events that do not implies

---

[1] JADE : http://jade.tilab.com/
[2] Protégé: http://protege.stanford.edu/
[3] OWL: http://www.w3.org/TR/owl-features/
[4] SWRL: http://www.w3.org/Submission/SWRL/

the behavior of devices, such as date, time, spatial consideration, human interactions and events, etc). For the reasoning itself, device collaboration rules are modeled using SWRL Rules. These rules are composed of two parts: the *antecedent* that represent the conditions to be evaluated and the *consequent* representing the result (the implications) of the rule when all conditions are verified. In our cases where they represent devices collaboration rules, the formalism is the following: the antecedent represent the conditions required to activate the consequent. Thus the consequent represent the action to be performed by the agent in term of resources

Fig 5 below depicts the overall architecture of the simulator, which is composed of three main aspects: multi-agent, business and data.

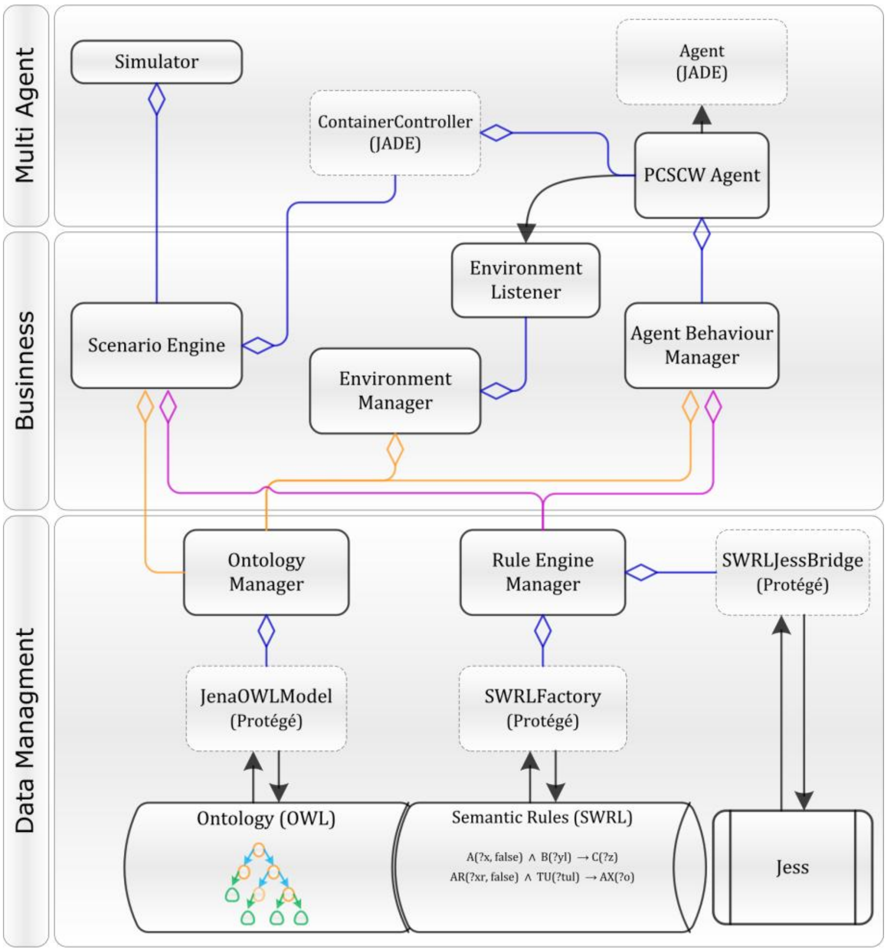

Fig 5. Simulator's Architecture.

On the top of the figure we've got the main object representing the simulator. Its role is to manage available scenarios and provide common features such as loading, saving, help and else. Bellow this main object we've got some vital modules:

Data aspect:

- *Ontology Manager*: this module is a "Facade" as it facilitates the use of the Protégé Framework by providing a set of simple methods specifically developed for the simulator. For instance it allows to efficiently retrieving all information (stored in OWL) about the initial environment of a scenario or about a specific agent.
- *Rule Engine Manager*: the rule engine manager is also a "Facade" but its role is a little different from the *ontology manager*. Indeed, while the *ontology manager* deals with OWL information retrieving and wrapping, the rule engine provides an interface not only for semantic rules (in our case SWRL Rules) but also to control and manage an SWRL Rule Engine which is in charge of running our semantic rules and thus trigger devices collaboration rules.

Business:

- *Scenario Engine*: responsible of the schedule and trigger of events in the simulation, it also has in charge to create the initial environment and manage the *ContainerController* of the JADE Framework that holds agents.
- *Environment Manager*: this module ensures the management of the Environment; it handles environment updates and dispatches environment events to agents.
- *Environment Listener*: this small module is used as an interface between the environment and agents; it provides simple methods to communicate environment's updates to agents and agents' updates to the environment.
- *Agent Behaviour Manager*: this is one of the most important modules of the simulator, it manages how the agent will react according to its context and preserve the coherence of its behaviour. For instance if an agent starts an action that may require several interactions with other agents, it has in charge to "memorize" and make the agent follow the process of the behaviour.

Multi-Agent:

- *PCSW Agent*: the base agent in our simulator, it inherits from the JADE *Agent* class and implements specific features to communicate with the environment and its *Agent Behaviour Manager* described above.
- *Simulator*: as already mentioned it is the main module of the application, managing the load of scenarios and eventually responsible for the management of one or several presentations.

5.2 Simulator's Streams

We have just detailed the technical architecture of our simulator; let's have a look at the way we feed it and how it provides us our useful results as it is presented on Fig 6.

The main input is obviously the scenario; it is composed of an initial environment storing all information about agents and other resources to represent the initial state of the scenario. In addition to this initial environment the scenario stores roles and tasks used by agents. In order to ensure the unfolding of the scenario we've got a set of events representing the different steps of the scenarios. Each event is triggered according to the evolution of the environment and agents. The second input managed by the simulator is device collaboration rules. As we want to make them as generic as possible we don't bound this rules to specific cases and we can then separate them from scenarios. We haven't represented them on the figure but it is obvious that we've got some other technical parameters for the simulation that do not directly interfere with the playing of scenario but with the user interface.

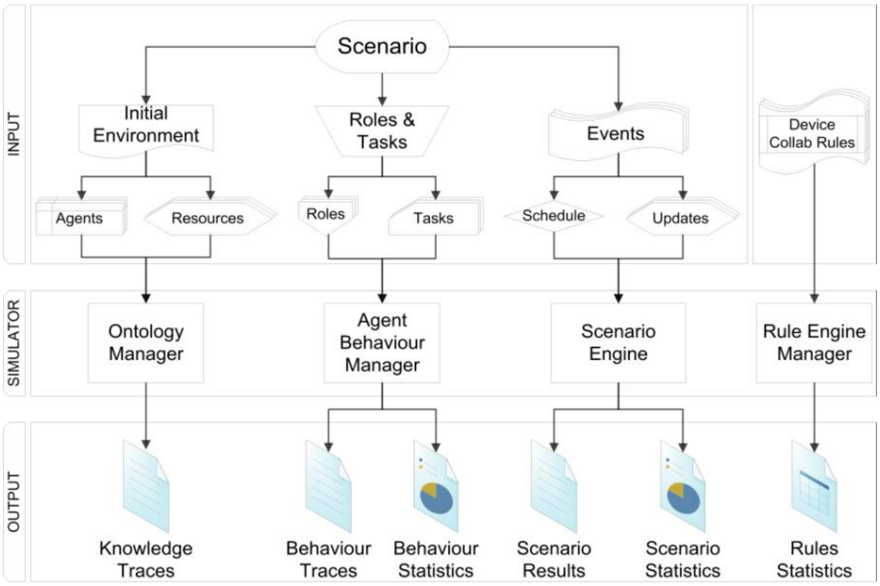

Fig 6 Simulator's Input and Output.

Once a scenario has reached its final step or its time limit it also has produced a set of outputs. The most obvious one is a report of scenario results, showing the execution time, the list of agents involved in the collaboration the last step reached and the state of the environment and agents at the end. We also produce some statistics concerning the scenario such as time to accomplish each task, the number of interactions between human and machines, any indicator that can help us analyze

the scenario. Concerning this last point we already have some trails [19] about the kind of indicator that could help us evaluate the collaboration in pervasive computing environments as we'll see in the following section. We also produce other kind of outputs: rules and behaviors statistics allowing us to evaluate the utility of devices collaboration rules. The last type of output consists in traces of behaviors and agents knowledge; combined with the original scenario they can be used to "replay" the past simulation. In addition this kind of output is particularly useful as it can help us finding way of improvement for device collaboration rules.
Let's see how these outputs can serve us to evaluate our work and the collaboration in pervasive computing environments.

6. Evaluation

Until now we have presented our new model for collaborative work in pervasive computing environments, the PCSCW model. In addition we are currently in the process of developing a simulator to validate this model. There is still a point that we haven't considered in this paper: how can we evaluate the model, its benefits and its drawbacks.

Classical computer supported collaborative work evaluation offers a large variety of methods. Thus, many different techniques have been used to evaluate groupware technologies, but for now no consensus has been reached on which methods are appropriate in which context. Existing CSCW evaluation methods can be organized according to two main categories: traditional methods applied to CSCW -the first tries to evaluate CSCW were done using single user methods- and methods especially created for it. Among these two main categories we can point out some similarities: *discount methods* aimed at providing low-cost evaluation, they offer a mean to quickly evaluate a system with limited cost and most of the time with few constraints but they have some limitations such as dissociation from the real work settings, lack of a real theoretical basis, weak coupling with the domain, lack of accuracy, *scenario-based methods* are effective in helping to focus evaluation efforts and in identifying the range of technical, human, organizational, and other contextual factors that impact system success. However, this kind of method is somewhat less useful for identifying the measurable benefits gained from a CSCW implementation due to the potential complexity to determine what part of the system has improved the collaboration. Finally we've got *task model based evaluation* aimed at evaluating the role of a user inside the collaboration, trigger some events at some point of the task, make statistics about the tasks, feed a context manager, but the applicability of this type of formal analysis is limited by the availability of quantitative data concerning the application, which in the case of collaborative software can be complex to collect and even more to interpret correctly.

As our work sticks a little out of the traditional perspective we had to find how we could evaluate our work. Such model cannot be fully evaluated theoretically; indeed to do it efficiently we need to evaluate its use. Then we base our evaluation on the analysis of the efficiency of the representation coupled to devices collaboration rules in the improvement of collaboration between humans.

Thus, one of the first things we needed to consider to conduct an evaluation of this model is how can we quantify benefits and drawbacks, how can they be measured, what kind of evaluation indicators can be used. To find those indicators we considered the real purpose of our model: simplifying the interactions between humans by automating some of the interactions that can take place between smart devices. From this perspective we can point out some concrete indicators:

- *Number of devices interactions to complete a task*: this measure has to be interpreted according to the objective of the collaboration, indeed users may want to limit exchanges between devices for cost or security reasons for example. Thus, a high number of interactions may mean that devices have well adapted their behaviour to help users, but also that they have overexposed users' documents.
- *Number of human interactions to complete a task*: this simple measure represents the number of human interactions (with devices of with humans) that has been required to perform a given task. For this indicator, fewer interactions often mean a simpler and more efficient collaboration.
- *Number of interacting humans / total number of humans*: this measure can indicate the complexity of the collaboration, the more humans there is to collaborate the more complex the collaboration will be. Then if there are fewer humans involved to perform a task it implies that our model facilitates the collaboration.
- *Time to complete a task*: this simple measure can have an important impact on the relevance of a device collaboration rule, indeed, even if a rule efficiently reduce the number of humans and devices interactions, it may not be usable if it dramatically increase the required time.

In order to facilitate the validation and the evaluation of our model we have designed our simulator to be able to simply handle this kind of measure. Thus, once completed we will have the opportunity to measure and compare the efficiency of collaboration with and without our model. This will also help us to improve our devices collaboration rules by providing us useful indicators.

7. Discussion

The model we propose does not come out of nowhere, it relies on robust researches that inspired us and guided us to develop it. As it has been intensively mentioned, our model is based on the notion of roles, for people and for devices. In a moral consideration, the representation of roles is not a substitution of the representation of a person, it is only a part of a person, otherwise one can quickly come to the conclusion that only roles matters and peoples don’t. But from a model perspective taking into account the role as a variable can help to apprehend the complexity of a pervasive environment. In such context, roles or resources can vary depending on spatial, temporal or collaborative constraints. Having a model in which the "efficiency" of the collaboration can be estimated may be used for designing purpose.

The PCSCW model is designed to facilitate the collaboration of users by making devices cooperate. In its nature this model could be considered as a meta-model as it tells how some sub models can be used and combined to improve the collaboration of users. Even though we can take out some benefits and drawbacks of our model put in regard with previous ones.

The most obvious benefit or our model compared with others is the fact that it natively considers the distribution of resources and the possibility to use them all at once. Indeed most of traditional collaboration models, based on tasks, roles or even more advanced collaboration awareness models focus on the way to keep users and their devices aware of the collaboration. We think the PCSCW model is going further in this direction by using collaboration awareness to enable the “collaboration intelligence” of devices and develop their proactive behaviors. Another noticeable and valuable benefit of our approach is the possibility to precisely monitor the current state of the collaboration. Indeed, as we have to depict each task and their related actions it helps channeling the collaboration awareness.

However we know that there are some drawbacks in our approach, the main one is probably the high level of description required by devices to adapt their behavior. Thus if we only need to have an overview of the collaborative activity some models (such as [12]) can propose faster, but less accurate, solutions than ours.

Indeed, our approach can seem very descriptive, detailed and requiring great efforts to be used. But we want to take an advantage from this issue. In fact, all awareness mechanisms do not require the same level of description. For some of them, only the top levels are relevant. This is why we argue our model is able to describe and reason on different granularity levels, from a simple description of devices until a fine description of each object manipulated by an application on a virtualized operating system. Thus, we can say that our model naturally supports the scalability of awareness mechanisms by its adaptability to the description of resources. This scalability can even bring an abstraction capacity by allowing designers to represent high-level information and reason on it. Besides, this scalability advantage is twofold, it allows the description of resources with various granularity, but it also offers the possibility to reason with few context information and then when computing resources are limited or information are hard to obtain.

This work relies on two main aspects: the representation of required and available resources and the description of device collaboration rules. Still we know our work has its own disadvantages. One of the mains is the need to create these rules. Indeed to adapt to a specific context it requires having a more or less generic set of rules. Even if this particular point can seem annoying it can be a source of im-

provement. Despite rules have to be written before the use of the model, they can also be derived from user's activity, preferences and constraints dynamically. As for now we don't have dig deeper in this way as it was not our main focus, but in future works we'll need to explore potential way to automatically generate and adapt rules.
We think our model is a good basis to develop interactions between smart devices. Besides the collaboration supported is threefold: collaboration between users, collaboration between devices and collaboration between users and devices.
Still, we know that our model certainly needs some improvements, and that without implementation and in-depth evaluation it is only theory. In this perspective the simulator we're currently developing will provide use a useful tool to validate and evaluate the efficiency of the model. Even if it is an essential tool for the evaluation process it is not sufficient for a real deployment. Indeed, if we refer to [14] and [15], the evaluation of a CSCW system has to be organized in three phases: laboratory evaluation without users to detect obvious problems, laboratory evaluation with a part of user's context and evaluation in real conditions to validate the scalability of the system and its deployed efficiency. Hence the next step for us after the development of the simulator and the laboratory evaluation will be to develop a prototype of the final application and make it evaluate by real users. There's no doubt this prototype will be very helpful to design new collaboration rules as users' feedbacks will go along. Combined together the simulator and the prototype will give us an efficient evaluation framework to create and improve devices collaboration rules in order to facilitate even more the collaboration between users.

References

1. Zhu, H. and Alkins, R.: a Tool for Role-Based Chatting. In: IEEE International Conference on Systems, Man and Cybernetics, pp. 3795—3800 (2007)
2. Zhu, H.: From WYSIWIS to WISINWIS: Role-Based Collaboration. In: IEEE International Conference on Systems, Man and Cybernetics, pp. 5441—5446 (2004)
3. Zhu, H., Tang, Y.: a Role-Based Hierarchical Group Collaborative Awareness Model and its Implementation. In: IEEE Information, Technologies and applications in Education, pp. 12-16 (2007)
4. Molina, A.I., Redondo, M. A., Ortega, M., Hoppe, U.: CIAM: A Methodology for the Development of Groupware User Interfaces. Journal of Universal Computer Science 14, 1435-1446 (2008)
5. Ahn, G.J., Zhang, L., Shin, D., Chu, B.: Authorization Management for Role-Based Collaboration. In: IEEE International Conference on Systems, Man and Cybernetics, pp. 4128-4134 (2003)
6. Barbuceanu, M., Gray, M., Mankovski, S.: Coordinating with Obligations. In: Agents '98, Minneapolis, May, 1998, pp. 62 - 69 (1998)
7. Edwards, W.K.: Policies and Roles in Collaborative Applications. In: Proceedings ACM Conference Computer Supported Cooperative Work, Cambridge, MA, pp. 11—20 (1996)

8. Hamadache, K., P. Manson, P., Lancieri, L.: Pervasive services, brainstorming in situation of mobility. In: 3rd International Conference on Pervasive Computing and Applications. Alexandria, Egypt, pp. 709-714 (2008)
9. Penichet, V., Lozano, M., Gallud, J.A., Tesoriero, R.: Task Modelling for Collaborative Systems. In: Task Models and Diagrams for User Interface Design, pp. 287-292, Springer Berlin / Heidelberg (2007)
10. Yan, L., Zeng, J.: A Task-Based Group Awareness Model. In: The 8th International Conference on Computer Supported Cooperative Work in Design Proceedings. Volume 2, 26-28 May 2004, pp. 90 – 94 (2004)
11. Vellis, G.: Model-based development of synchronous collaborative user interfaces. In: Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems. Pittsburgh, PA, USA, (2009)
12. Rodden, T.: Populating the application: A model of awareness for cooperative applications. In: Proceedings of the ACM CSCW'96 Conference on Computer Supported Cooperative Work, pp. 87-96. ACM Press, Boston, MA (1996)
13. Zhu, H., Seguin, P.: The Role Transition Mechanisms in Role-based Collaborative Systems. In: Canadian Conference on Electrical and Computer Engineering, pp. 1359—1362 (2005)
14. Herskovic, V., Pino, J.A., Ochoa, S.F., Antunes, P.: Evaluation Methods for Groupware Systems. In: J.M. Haake, S.F. Ochoa, and A. Cechich (Eds.): CRIWG 2007, LNCS 4715, pp. 328–336. © Springer-Verlag Berlin Heidelberg. (2007)
15. Hamadache, K. and Lancieri, L.: Strategies and Taxonomy, Tailoring your CSCW Evaluation. In: Nelson Baloian and Benjamim Fonseca (Eds): CRIWG 2009, LNCS 5784, pp. 206-221 © Springer-Verlag Berlin Heidelberg. (2009)
16. Mori, G., Paternò, F., Santoro, C.: CTTE: Support for Developing and Analyzing Task Models for Interactive System Design. IEEE Trans. Softw. Eng. 28(8): 797- 813 (2002)
17. Drury, J., Williams, M. G.: A framework for role-based specification and evaluation of awareness support in synchronous collaborative applications. In: Proceedings of the Eleventh IEEE International Workshops on Enabling Technologies, pp. 12-17. IEEE Press, Pittsburgh (2002)
18. Gutwin, C., Greenberg, S.: A descriptive framework of work space awareness for real-time groupware. Computer-Supported Cooperative Work, pp. 411-446 (2002)
19. Hamadache, K., Lancieri, L.: Dealing with Device Collaboration Rules for the PCSCW Model. In: 16th Collaboration Researchers' International Working Groupware Conference on Collaboration and Technology (CRIWG 2010), September, Maastricht, the Netherlands (2010)
20. Linxia, Y.: realization of task-based group awareness model. In: International Conference on Computational Intelligence and Software Engineering, pp.1 – 3 (2009)