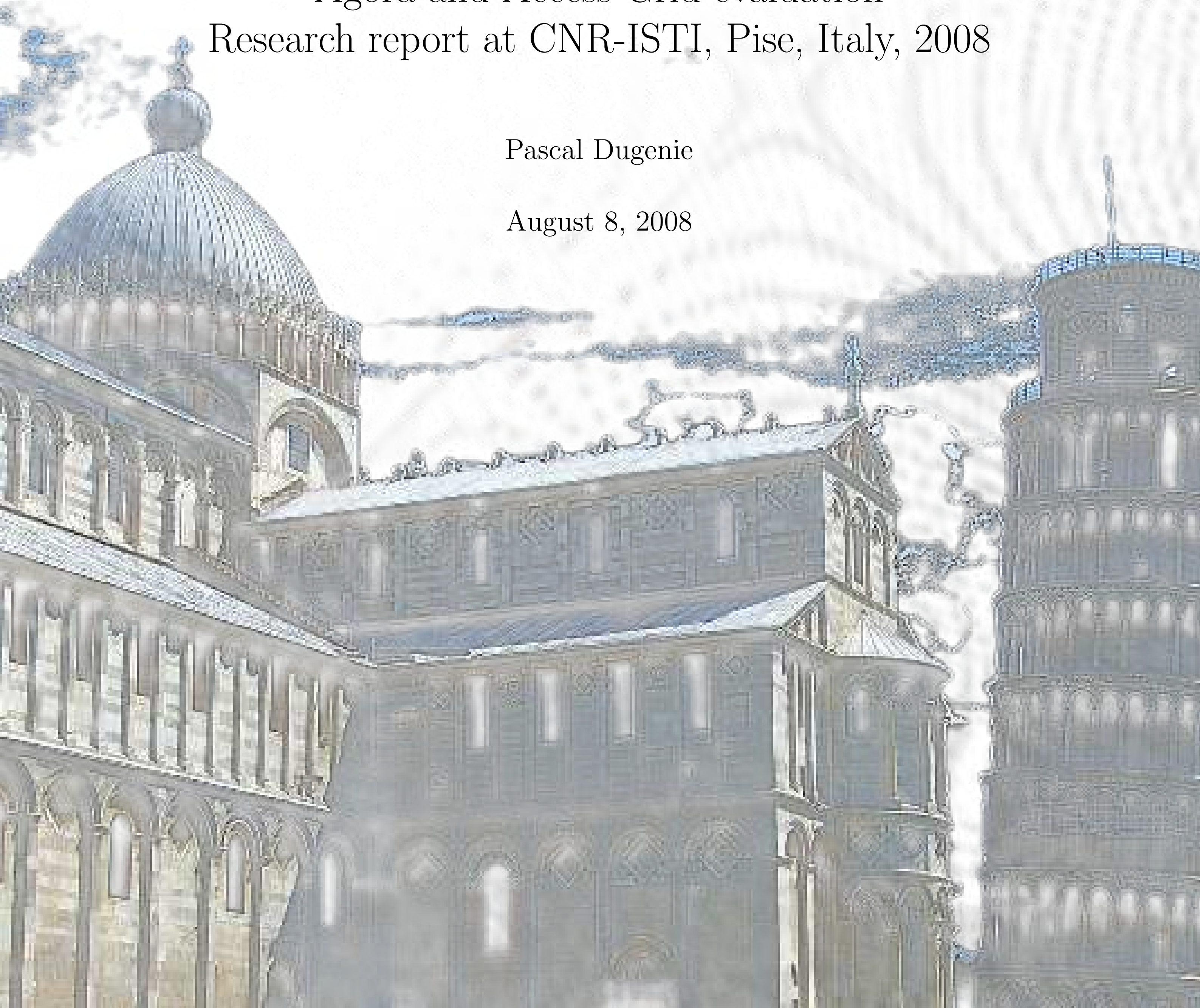
Agora and Access Grid evaluation at CNR-ISTI, Pise, Italy, 2008

Pascal Dugénie

Agora and Access Grid evaluation
Research report at CNR-ISTI, Pise, Italy, 2008

Pascal Dugenie

August 8, 2008



Summary

Abstract

This report relates a six months activity at the ISTI-CNR in Pise, Italy. The ISTI and the LIRMM seize the opportunity to develop a partnership on the common interest in developping a service oriented collaborative environment platform on a GRID infrastructure. This particular activity consists to evaluate some technical aspects of the platform AGORA UCS developped by the LIRMM and Access Grid (AG) a widely used collaborative environement of the GRID.

Chapter 1

Introduction

1.1 Context

Pascal Dugénie, ingénieur CNRS, spent 6 months at ISTI-CNR, Pisa, Italy, from March 2008 to evaluate some technical aspects of the platform Agora UCS and Access Grid (AG). This activity is a joint initiative between two european research laboratories. The ISTI[1] and the LIRMM[2] (Montpellier, France) who seize the opportunity to develop a partnership on the common interest in developping a service oriented collaborative environment platform on a Grid infrastructure. This has been made possible thanks to funds from the CNRS strongly encouraged by Prof. Stefano Cerri and the CNR supported by Dr Domenico Laforenza. We acknowledged gratefully Dr Domenico Laforenza, who was at that time head of the HPC (High Performance Computing) laboratory for his kind support to enable this initiative. We congratulate Dr Laforenza who has been appointed on 1st July as Director of the IIS (Istituto di Informatica e Telematica) a large institute of the CNR based in Pisa.

Figure 1.1: Dr. Pascal Dugénie

[1]ISTI (Istituto di Scienza e Tecnologie dell'Informatione) is an institute of the CNR (Consiglio Nazionale delle Ricerche), which was constituted in September 2000 as a result of a merger between the IEI (Istituto di Elaborazione dell'Informazione) and the CNUCE (Centro Nazionale Universitario di Calcolo Elettronico exists since 1965 and became part of the CNR in 1974). More info is available at www.isti.cnr.it.

[2]LIRMM: www.lirmm.fr

Figure 1.2: Dr. Domenico Laforenza

Figure 1.3: Prof. Stefano A. Cerri

1.1.1 About the HPC laboratory

HPC (High Performance Computing) is the welcoming laboratory within the ISTI. It is currently composed of 15 members including 9 permanent research staff. Raffaele Perego is the new head of HPC lab since the 1st July (see photo of the team on figure 1.1.1).

HPC is involved in six european FP6 projects including XtreemOS and CoreGrid, and one FP7 project called S-Cube. There are also two national projects including Escogitare[3] funded by the Ministry of Agricultural and Forestry Policy (MIPAF) which aims to provide an infrastructure to the Italian Agricultural REsearch Council (CRA) in order to enable the different institutes to share their resource and knowledge. All these projects are mainly focused on GRID environment.

[3]Escogitare (Un'infrastruttura di E-SCience per gli Instituti di Ricerca del aGroalimenTARE (2004-2008).

Figure 1.4: People of the HPC lab and associates

1.2 Objectives

This activity aims to evaluate various aspects of two distributed collaborative environements in order to determine the potential for interoperability and performance issues. The ISTI has deployed an AG node and the LIRMM has developped the Agora UCS service. Both institutes are interested in exploring the domain of inter-operaterability between AG and Agora UCS services. Both environments provide remote meeting facilities such as shared visualisation and audio-video conferencing. During the period of this activity, three objectives have been envisaged:

- deploying a new Agora UCS node at the ISTI;
- evaluating Agora UCS and AG performances and potential for interoperability;
- disseminating the knowledge gained by this activity; this is done by the present report and four articles that have been submited in international conferences.

1.3 Methodology

The methodology adopted along this study may be decomposed into three steps:

- The first step was to deploy an Agora UCS node at ISTI and begin to mutualise resources of both ISTI's and LIRMM's nodes.

- The second step was to investigate technical aspects of both AG and AGORA UCS architectural models, in order to determine theoretical level of inter-operability and identify possible alternative solutions.
- The third step was to analyse the performances of both systems, in order to determine some practical limits.

Chapter 2

Background

2.1 Collaborative environments

2.1.1 Virtual Collaborative Environments

A Virtual Collaborative Environment (VCE) is a concept mainly studied in the domain of CSCW[1]. A VCE is a set of distributed computing resources, services and interfaces aiming to provide an advanced environment for collaboration. These interfaces may support several modalities of communication such as audio-video, shared visualisation, instant messagging, notification and shared file repositories.

2.1.2 VCE Principles

Three principles govern the concept of VCE. They result from a top-down approach in the analysis of the problem of setting up a collaborative environment when resources and agents are distributed. Both static and dynamic analyses can be considered separatively. The static analysis relates to a *model of organisation* of the entities within the collaborative environment (ie. the resources and the agents). The dynamic analysis relates to a *model of activity* of these entities within the VCE (ie. interaction and communication between the agents).

Ubiquity This principle has been envisioned more than a decade ago in informatics to express a way to access to computing resources from anywhere at anytime [26]. Nevertheless, concrete deployment of operational solutions has become feasible only recently by means of pervasive technologies such as Grid. This principle impacts both the organisation and the activity models of the VCE architecture. Concerning the organisation model, the principle of ubiquity stipulates that computing resource provisionning required by the collaborative environement must be entirely delegated to the infrastructure. This implies that this infrastructure must be dimensionned to provide distributed storage and

[1] Computer Supported Collaborative Working

processing capacity with a suitable connectivity to the end-user. Concerning the activity model, the principle of ubiquity requires resource able to maintain the state of the interactions and communications between agents. However, the state of these interactions must be independant from the physical location of these agents. This enables to adopt thin terminal equipment with a basic connectivity and delegate the transport of large data flow to the infrastructure.

Awareness This principle, which inherits from many years of research in CSCW [3], covers various aspects of communication to share knowledge about the activities of other members of a collaborative environment. Typically, CSCW claims that the use of multiples modes and interfaces of communication (*i.e* multimodal communication) reinforce the sentiment of presence, hence greatly influence the motivation factor. Therefore, increasing the *awareness* improves the efficiency of the collaboration process. These collaborative services fall into one of these two categories: synchronous (same time) or asynchronous (not depending on others to be around at the same time). Synchronous services may include audio-video conferencing, instant messaging and shared visualisation. Asynchronous services include collaborative editing tools like wiki and version control systems. The notion of *awareness* between humans can be seen like the notion of coupling between computing resources.

Immanence This principle refers to philosophical and metaphysical concepts. However, immanence expresses, in a wider sense, the idea of a strong interdependance between the *organisation* and the *activity* of a complex system (see 2.1.2). An immanent system is a system that constantly re-constructs its own structure throughout its internal activity. In other words, *The organisation is immanent to the activity.* By opposition, a system whose behaviour would be completely determined from the initial conditions with no feedback effect of its activity on its own structure is not an immanent sytem and has no chance to be self adaptive in case of changes of conditions of its environment. Examples of immanent systems are living systems in biology or social organisations. The principle of immanence has been introduced in informatics to describe the social impact derived from the introduction of the internet in the society. Furthermore, the expression *collective intelligence* has been adopted to describe the immanent system of knowledge structured around the Web [22]. The material (*i.e* the Web) is immanent to the immaterial (*i.e* the collective intelligence). Since the principle of immanence constitutes the *living link* between the organisation (*i.e* the static model) and the activity (*i.e:* the dynamic model) of a system, both models act upon each other. The organisation enables to generate the activity whereas the activity constantly seeks to improve the organisation.

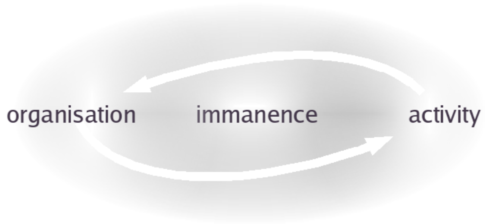


Figure 2.1: Principle of immanence

2.1.3 Immanence in collaborative environments

The notion of immanence, which was appearing quite utopic only a few years ago, is gaining an increasing interest in informatics because of technological maturity to demonstrate its faisibility. Also, this principle is becoming highly critical in the analysis of complex problems. Thus, there are many reasons for considering the immanence in GRID collaborative environments.

One reason is the possibility offered by the GRID infrastructure to deploy autonomous services. These services can be instanciated somewhere in the infrastructure, and adopt a proactive behaviour. This is a major difference between the GRID over the Web which is not able to provide stateful resources necessary to operate autonomous services.

Another reason is the trend for the holistic approach for modelling collective behaviour in Multi-Agent Systems (MAS). In this approach, interactions between agents are contextualised within a global collaborative process. The notion of agent is extended to cover artificial processes as well as human ones [2, 18]. Agents interact within a collaborative environment by providing or using services. One essential condition for a collaborative environment to become immanent is that any agent of the system may play an active role in the system construction [10, 4, 5, 8]. For instance, both system designers and expert-users have a symmetrical feedback in the cycle of developing and validating a complex application. They interact by providing services to each other via a common collaboration kernel. They may develop their point of view in the context of a collaboration process and their role may evolve indefinitely. Thus, such a system clearly requires self-adaptiveness and self-organisation.

2.1.4 The use of Grid for VCE

The interest of a GRID infrastructure for a VCE is to allows a seamless access to distributed computing resources. GRID contributes mainly to achieve the principle of *ubiquity* since GRID resources are able to maintain the state of the communications within the infrastructure and independently from the location of the terminal elements.

2.2 Virtual organisations

2.2.1 VO management models

In its original definition, a VO is a community of users and a collection of virtual resources that form a coherent entity with its own policies of management and security [16]. A rudimentary VO management system has been originally built-in *Globus* but has little potential for scalability. In order to resolve these limitations several VO management models have been proposed within the GRID community.

Community Authorization Service (CAS) has been specifically designed to facilitate the management of large VO [23]. The functionalities for VO membership and rights management are centralised in a LDAP[2] directory. Since, the structure of the VO is strongly hierarchichal, this is hardly possible to reorganise the initial tree once the services are deployed.

VO Membership Service (VOMS) [24] is deployed in more recent GRID infrastructures such as EGEE. It resolves some problems of CAS such as the membership management by providing a more evolutive relational database instead of a flat tree structure. However, VOMS still presents some conceptual limitations such as an inheritance link between a parent VO and its childrens. The subdivision of VO into groups often creates confusion in the management of rights and does not enable a complete independence between the groups and the VO. For instance, the lifetime of a group is determined by the lifetime of the parent VO.

[2]Lightweight Directory Access Protocol

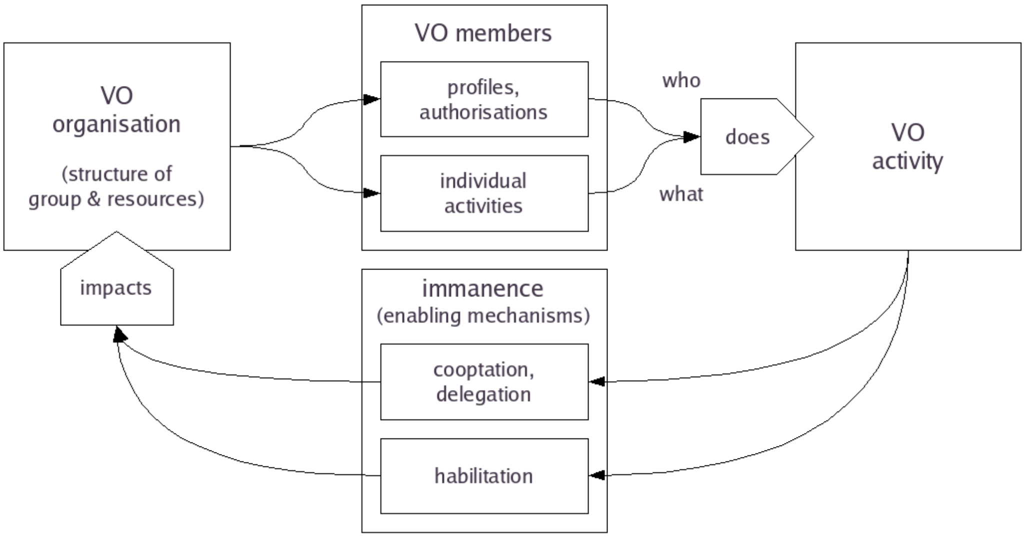


Figure 2.2: A typical VO management model

2.3 Discussion

The research activity in Grid eludes the question of immanence because two complementary aspects are usually treated separately while they should be treated together. The aspect of *collaboration* in general is treated in domain of CSCW[3]. The aspect of *VO management* is studied through various conceptual models of organisation.

2.3.1 A typical VO management model

Designing an architecture allowing access to the resources of a VCE is a real technological challenge. Indeed, the models presented here are based on client-server architecture with several points of rigidity. It results that the end-user may face usability constraints related to technological choices adopted more or less arbitrarily by the designer of the architecture.

AG focusses on the principles of *awareness* and *ubiquity*. However, AG does not include a powerful mean for VO management. VO are managed in an *ad hoc* manner at the venue server side. This has the inconvenience to require much technical administrative work from computer experts in this domain. Therefore, AG has no potential for immanence.

Figure 2.3.1 represents a typical VO management system. It illustrates why both CAS and VOMS fail in ensuring self-organisation of VOs. On

[3]CSCW: Computer Suported Collaborative Work

the left part, the VO management system determines the overall system organisation. On the middle part, mechanisms for VO member management and, on the right part, the resulting activity is directly dependent on the initial organisation. At this stage, there is no more possibility to re-introduce activities back to the system organisation.

The bottom part of the figure represents the mechanisms for enabling the immanence principle. It includes processes such as *cooptation* (a set of protocols to introduce new members), right *delegation* between VO members and *habilitation* to perform tasks in the context of the VO. This involves many kinds of knowledge transfer mechanisms that are ensured during the collaboration activity.

2.3.2 Grid and MAS convergence

The convergence of GRID and MAS research activities brings up new perspectives towards a immanent system. Firstly, GRID SOA[4] is now established as a powerful framework for resource and security management. Moreover, a SOA enables to abstract the underlying technology behind the unique concept of *service* [15]. Secondly, MAS focusses on complex organisation models far more advanced than the GRID ones. In particular, the MAS conceptual model Agent-Group-Role (AGR) presents the characteristics of flexibility required for a self-organised organisation model [12].

Although, the integration of GRID SOA and MAS concepts has been proposed to enhance the potential of organising distributed resources and agents [14], this integration has required an extensive effort of formalisation with AGIL, the Agent-GRID Integration Language, that has been settled only a couple of years ago [11, 20, 19]. This formalisation has been used to abstract GRID and MAS concepts and develop new kind of architectures. AGORA formed with both GRID and AGR concepts is an example of such an architecture.

[4]An SOA (Service-Oriented Architecture) has been adopted in the Open GRID Service Architecture that has become the reference model for GRID systems

Chapter 3

Agora and Access Grid

Agora and Access Grid (AG) are two kinds of VCE. AG appeared consecutively to the emergence of Grid technologies and now benefits from a widely deployed infrastructure accross the world. Agora has been very recently designed as an experimental platform. AG benefits from a large adhesion accross the world, but, according to the latest experiments, Agora present a strong potential to facilitate the access to large computing resources.

3.1 About Agora

Agora is an original VO model which exhibits the principles of *immanence*, *ubiquity* and *awareness*. Moreover, for experimental purposes, the VCE platform called Agora Ubiquitous Collaborative Space (UCS) has been developped four years ago in the context of the european project ELeGI[1] when the participants could not identify a VCE on Grid that minimize the number of intervention of software specialists. In order to demonstrate the effectiveness of the solution, extensive experiments of the Agora UCS prototype have been performed with more than eighty users accross the world [4].

3.1.1 Conceptual model

The conceptual model of Agora UCS presented on figure 3.1.1 consists of a set of **five concepts** and **four relations**.

Agent This concept constitutes the active element within the system. Every action of an agent modify the state of the system. An agent may be a artificial process but may also be a human.

Group This concept is a set of agents who are considered as members of this group. The term *community* may also be used to quality more accurately this kind of group. A group can be considered as the context where agents exerce their activity. An agent may be a member of several groups and exerce a different activity according to the group.

[1]ELeGI (the European Learning Grid Infrastructure), 2004-2007, www.elegi.org

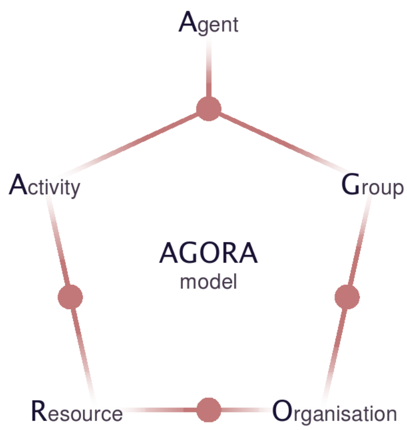


Figure 3.1: AGORA conceptual model

Organisation This concept is the general name which describe a formation of one given group and one unique set of resource. In the case of a distributed computing infrastructure such as GRID, this concept corresponds to the VO.

Resource This concept is a set of means to carry out tasks. In the case of a distributed computing infrastructure such as GRID, this concept corresponds to a *service container*[2].

Activity . This concept describes the way services are exchanged in the context of a group. It involves the notions of role, rights and interaction between agents.

The four relations consist of one ternary relation and three binary relations.

Relation agent-group-activity expresses that an agent may become member of one or several groups and exerce different activities in the context of one of these groups. This ternary relation indicates that a particular activity is specified for a couple agent-group. In other words, a set of activities are carried out by members of a group. The activity of a group can formally be defined as the sum of activities of this group.

[2] *i.e* a GRID service container can be defined as the reification of a portion of computing resources that have been previousely virtualised through a GRID middleware.

However, the effect holistic exercised by the group on members does not confirm this equality.

Relation activity-resource indicates that any activity requires the presence of resource. This can be running a process, accessing to storage facilities or peripherials.

Relation group-organization is bijective and indicates that any group is attached to a unique organization. In other words a group is a part of an organization.

Relation resource-organization is bijective and indicates that a set of resource is attached to a unique organization. Like a group, a set of resource form a part of an organization.

The ternary relation between the three concepts, *agent*, *group* and *activity*, enables to resolve some limitation of existing VO management models. For example, assuming that two agents are members of the two same groups, their activity may be completely asymetrical according to the group in the context of which they are considering to collaborate. In existing VO management solutions, based on a hierachichal model, this asymetrical relationship would not be possible since two agents would have a unique level of interaction.

3.1.2 Persistent Core Services

Agora UCS model includes a number of six persistant core services necessary for bootstaping and maintaining a collaborative environment.

Figure 3.1.2 is a representation based on the AGIL formalism, that shows a service coontainer including six PCS:

1. **A** **uthorisations**: Members of a VO may have a different level of permission on services. This service is in charge of assigning rights to members including the permissions over the PCS.

2. **M** **embers**: A VO is composed of members. This PCS manages the description of members of a VO, adding or removing members.

3. **G** **roup**: A VO is characterised by its properties (identifier, description, etc.). Also, the creation of a new VO is always performed in the context of another VO. Therefore this PCS is in charge of both intra VO operations as well as extra VO operations.

4. $\boxed{\mathbf{H}}$ **istory**: All the data belonging to a VO must be stored, maintained and also indexed. This PCS is in charge of keeping track of changes, logs of events and also of recording collaboration sessions.

5. $\boxed{\mathbf{E}}$ **nvironment**: A VO may personalise its environment. This environment operates in a service container. This PCS is in charge of adding or removing services (excluding the PCS).

6. $\boxed{\mathbf{N}}$ **otifications**: Communication between members of a VO and services is performed via notifications. This service treats the flow of notifications and manages the states of the exchanged messages.

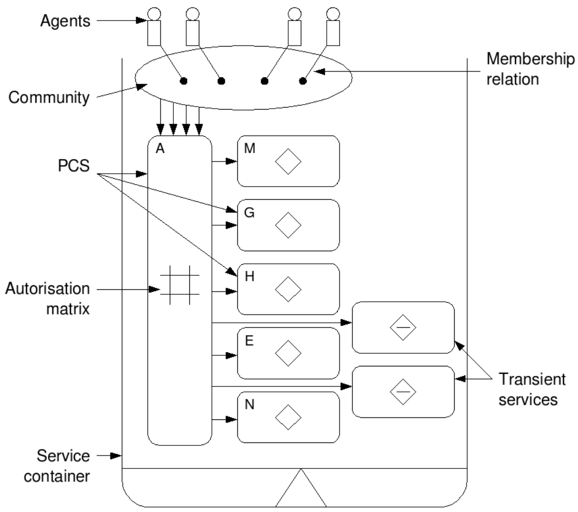


Figure 3.2: The six PCS

Since one service container is associated to a VO, there are as many sets of PCS as VO.

3.2 About Access Grid

Access Grid[3] (AG)is the most world-wide deployed GRID VCE. AG operates on *Globus* [13], the most popular GRID middleware. The topology of the Access Grid infrastructure consists of two kinds of nodes: the venue clients and the venue servers [25]. AG venue clients can meet in a venue server to set up a meeting. AG uses the H.263 protocol [17] for audio and video encoding

[3]Access Grid: www.accessgrid.org

and multicast method to distribute the communication flow between sites. The display of multiple H.263 cameras in every site gives a strong feeling of presence from every other site. The modular characteristic of AG allows to add new features such as application sharing (shared desktop, presentation, etc.) and data sharing. AG focusses on the principles of *awareness* and *ubiquity*. However, AG does not include a powerful mean for VO management. VO are managed in an *ad hoc* manner at the venue server side. This has the inconvenience to require much technical administrative work from computer experts in this domain. Therefore, Access Grid has no potential for immanence.

AG documentation AG technical specifications are described in a series of documents:

- The quick installation guide [1]
- The AG3 venue client user manual [25]

3.2.1 A quick comparison

At first glance Agora UCS and AG seem to offer similar functionalities. In particular, the shared visualisation coupled with audio-video conferencing to offer some more modalities of communication. However, both address a different target. While AG focuses on largely equipemed meeting rooms with multimedia facilities, Agora UCS focuses on simple access from a basic user terminal (typically a web browser, a thin client or a mobile terminal equiped with a webcam). In fact Agora UCS does not need to install specific softwares since the usual plugins (flash, java) are built-in most client configurations. For this reason, Agora UCS services instances delegates most processing at the infrastructure side.

AG has more advanced functionalities for session management, in particular the indexing and the retrieval of recorded sessions with AG video recorder. The implementation of session recording has not been achieved yet Agora UCS but it is becoming now a priority.

AG follows a Global Quality Assurance Program (GQAP) that gives more confidence in using it without risking problems of unstability. However, AG present some incompatibilities between versions 2 and 3. Assessment reports several porting errors between X11, Vic, Rat MSWin and Mac pdf formats. Even without GQAP, Agora UCS is little affected by this kind of risks of incompatibilities between releases since there is no cross platform communication done directly at the client side.

Topologies

Another major difference between AG and AGORA UCS architecture is the topology. The AG topology is composed of a set of venue servers and AG nodes that correspond to the end-user access point. The resource provision is distributed in both venue servers that provide the main resource and via multicast switches. AGORA UCS delegates all resource management at the infrastructure side. The client side is stateless which means that all information required for the communication is carried by the flow exchanged between the client and the server.

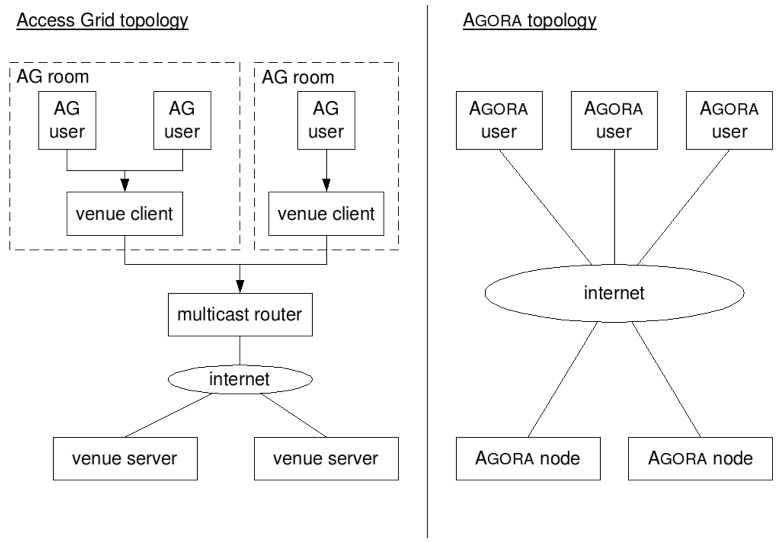


Figure 3.3: Topologies of AG and Agora

3.3 Trials and evaluations

3.3.1 Trials within the Escogitare project

Experiementations of AGORA and AG have been conducted between ISTI, in presence of Danele Bocci and myself at Pise and Mauro Magini, located at Rome, in the context of the Escogitare project [21]. Two sessions have permitted to evaluate the performances of AGORA and AG. The set-up of AGORA was immediate for Mauro Magini. However, we have had difficulties to synchronise the sound between sites. It took every time more than 30 minutes to be perfectely operational.

3.3.2 Other trials

We have performed three trials of AG alone in collaboration with the department of oceanography of the CNR. The first trial was done between Pise and the AG node of Southamton. In the second and the third meeting participants from other places joined the venue along with Pise and Southamton.

3.3.3 AG evaluation

Everytime, the time to set the AG environment exceeded 40 minutes for at least one site. The need for technical assistance was important since the parametrisation requires high knowledge in encoding and transmission techniques. Often problems were due to multicast problems because multicast is not always available on all sites. Bridges multi to unicast are often needed.

Real situation shows that there is a strong need for technical support to set a meeting. We observed a typical period of 20 mins on the phone before starting a session with the right communication parameters. During a session, there has been three lost of video connection in one hour. The solution is to exit and re-enter the venue.

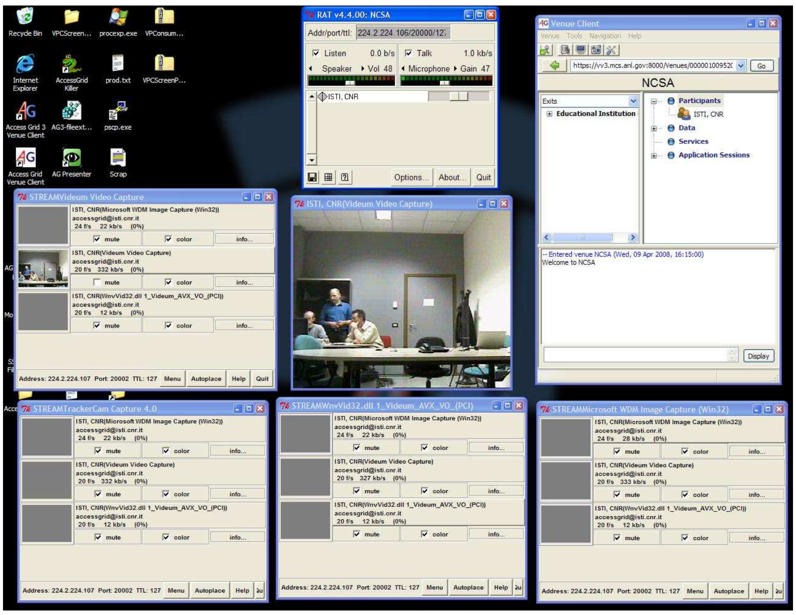


Figure 3.4: AG screen shot

3.3.4 Traffic profiles and routes

The figure 3.3.4 represents the traffic profile of a H.263 flow in AG. The black line corresponds to the downlink traffic while the red line corresponds to the uplink traffic. The average bandwidth used for this service is less than 100 kB/s with a peak of about 160 kB/s. Previous experiments of AGORA have estimated the traffic of the shared desktop to be about this range of bandwidth [4]. The figure 3.3.4 represents the routes used by the packets between Pise and Montpellier.

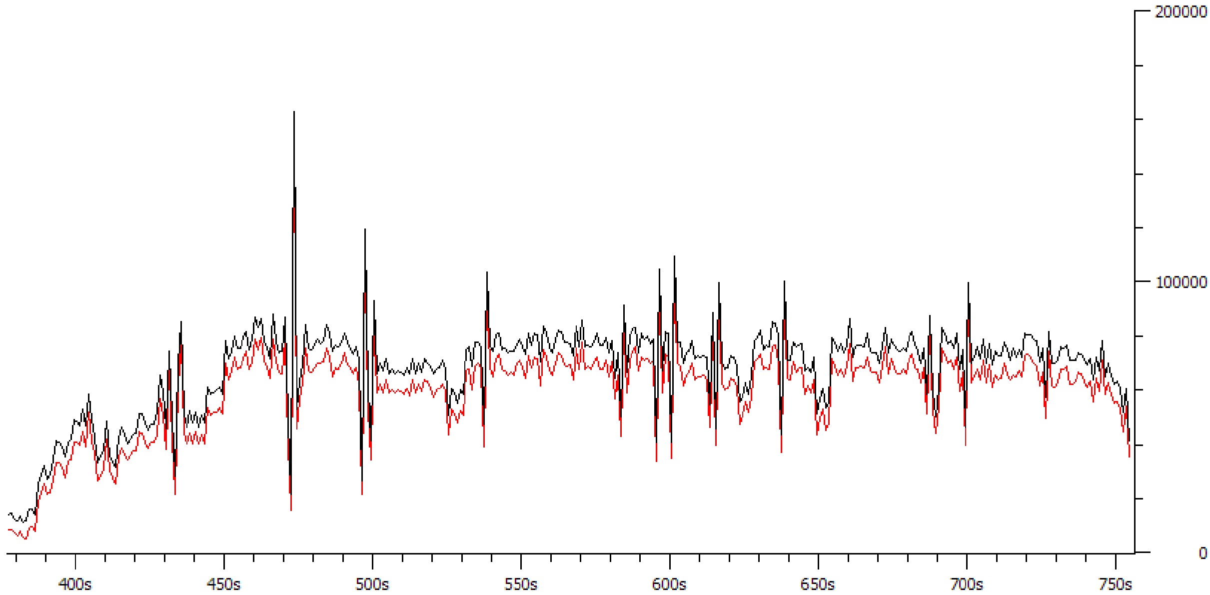


Figure 3.5: Traffic profiles

3.3.5 Experimentations of Agora

Extensive experiments have been done during the past four years. These experiments allowed to validate the mechanisms for immanence by focusing on user self-ability to feel at ease in AGORA UCS. Only a simple web browser acting as a *thin terminal* is necessary. The users often noted as important the ubiquitous access to the collaborative environment with no resource provided from their part. This allowed an immediate bootstrap of new VO and the acceptance of the technology was extremely high. The strong level of awareness allowed by the shared visualisation enabled a fast transfer of knowlege in particular for mastering complex computational tools.

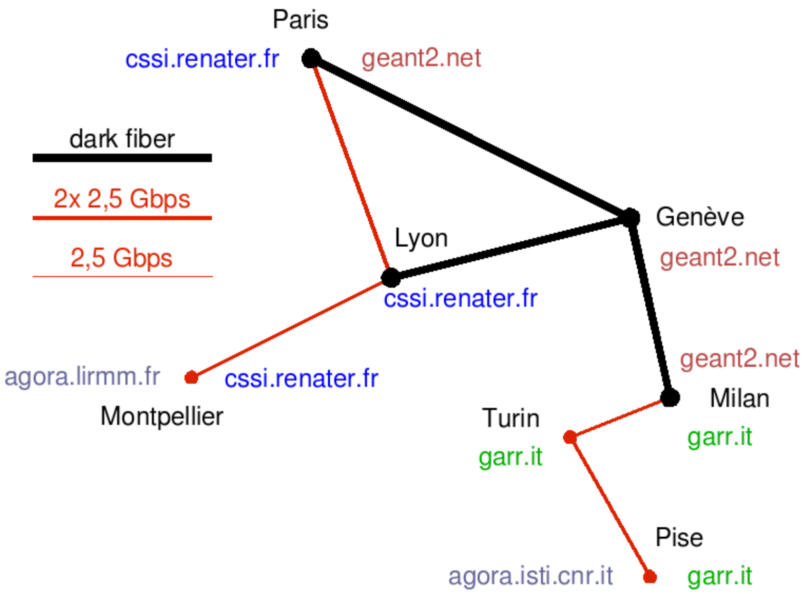


Figure 3.6: Main routes between Agora nodes

For instance a scenario called EnCOrE[4] has provided the most relevant results. Agora UCS enabled the visual representation of chemistry models at a distance. Most attention was put by the users on the semantics of their domain rather than solving computing problems. Unskilled users were at ease in their operations. The delegation of rights was important in the absence of some members. The cooptation of new members was also necessary to build a trustful community.

Since the behavior of a VO can not be forseen in advance, the flexibility of the Agora UCS is essential to enable a community to freely organise itself. Various situations of collaboration with reinforced modalities of interaction by using a synchronous communication interface have favored the transfer of knowledge. Discussions in real time, combined with visual representations on a shared desktop, allowed the actors to increase the effectiveness of the collaboration process.

[4]EnCOrE: Encyclopédie de Chimie Organique Electronique.
Demonstration available at http://agora.lirmm.fr

Chapter 4

Achievements

4.1 Agora prototype

Although some experiments to evaluate the potential of inteworking between Agora and AG have been done, most of the activity has finally been focused on improving the Agora prototype.

4.1.1 Refinement of Agora architecture

Some aspects of the Agora architecture have been refined. In particular the meaning of the two main concepts of the architecture model have been revised. The concept of *activity* is now extended to embrace notions of interaction, communication, interfaces, service, authorisations and roles. The concept of *organisation* specifies the association of one given *group* (and only one) and one set of *resource* (and only one) assigned to the activity of this particular *group*.

4.1.2 Optimisation of the code

Several parts of the source code have been revised to improve the overall performances. This concerns in particular the mechanism of serialisation of the access to the collaborative environment. Rather than using a single queue for all accesses, two queues are used instead. The first queue handles the incoming connection requests that are asynchronous for evry members of a group. The second queue handles the access to the diffferent groups of one particular user, once this user has been authenticated.

4.1.3 Synchronisation of the Agora web sites

There are now three web sites for AGORA services that are synchronised:

- http://agora.sourceforge.net
- http://agora.lirmm.fr
- http://agora.isti.cnr.it

4.1.4 Agora tree structure and packaging

The AGORA tree structure has been refined in order to facilitate the packaging and deployment. However this aspect requires still much work to be completely self installing.

4.1.5 Automation of the deployment

Several scripts to facilitate the deployment of AGORA nodes have been developped. However, this part is strongly linked to the previous activity, therefore there is still much validation process to achived completely a self-deployable automation mechanism.

4.2 Dissemination

The dissemination has been done mainly through the submission of articles in various conferences. The focus has been put of the description of the AGORA architecture in respect to the state of the art, including AG. The selected conferences are:

ITS (Intelligent Tutoring Systems) is the nineth conference held in Montreal.

ECAP (European conference on Computing And Philosophy) was held on the Campus of the University for Science and Technology, Montpellier.

SASO (IEEE International Conference on Self-Adaptive and Self-Organizing Systems), the second edition will be held in Isola di San Servolo (Venice), October 20-24, 2008

AWESOME will be held in Mexico.

HPDC (High-Performance Distributed Computing) was held in Boston. This conference combines the advances in research and technologies in high speed networks, software, distributed computing and parallel processing to deliver high-performance, large-scale and cost-effective computational, storage and communication capabilities to a wide range of applications.

The rate of acceptation is not as high as wished, in particular the reject from HPDC is disapointing. However, three or four articles in workshops will have been accepted for this six months period which is quite satisfactory in that respect.

Conference:	Author(s)	Status
ITS'08 [8]	P. Dugenie and S.A Cerri	In lecture notes
	P. Dugenie	Demonstration
ECAP'08 [5]	P. Dugenie	Accepted
SASO'08 [7]	P. Dugenie, and S.A Cerri	In review
AWESOME'08 [9]	P. Dugenie, C. Jonquet and S.A Cerri	In review
HPDC'08 [6]	P. Dugenie, and S.A Cerri	Rejected

4.2.1 ITS'08 workshop

Agora UCS
Ubiquitous Collaborative Space

Pascal Dugénie, Stefano A. Cerri, Philippe Lemoisson, and Abdelkader Gouaich

LIRMM: Université Montpellier 2 & CNRS; 161 rue Ada, Montpellier, France

Abstract. Agora UCS is a new architecture designed for distributed learning as a side effect of communication and collaboration. This architecture aims to achieve (i) ubiquity (time and space independent access by community members); (ii) immanence (full internal control of the destiny of the community) and (iii) multi-modal communication (reinforcing the interactions between the members of the community). The theoretical model underlying Agora UCS is inspired by an integration of agents and Grid concepts (AGIL). Agora UCS has been experimented by a dozen of communities, which represent alltogether about seventy members. We achieved quite promising results in terms of motivation and collective performances[1].

1 Requirements

Conversation and collaboration is the foundation of human learning. We base our requirements on a learning theory consisting of conversational cycles [9]. This theoretical background led us to draft the essential characteristics of a collaborative environment supporting mutual understanding and joint work: (i) immediate awareness of the life of the community; (ii) ability to maintain a clear and unambiguous internal model of the working environment and ongoing processes during the successive collaboration sessions. These general ideas have grounded the concept of Agora UCS in order to: (i) allow secure access to services in a terminal independent way; (ii) support ubiquity yet keeping the state of the collaboration independently from the user access location; (iii) dynamically allocate resources for any use of any service within a pool of mutualized and virtualized physical resources.

2 Architecture

The integration of Agents [1,2,3,7] and Grid concepts [5,6] has been extensively modeled in [8]. Figure 1 represents the Agora UCS architecture which is based on a ternary relation between three concepts. The *VO (Virtual Organisation)*, a concept borrowed from Grid, is a *community* of *agents* associated with dedicated

[1] As partners from the European ELeGI project (European Learning Grid Infrastructure, IST-002205): Joost Breuker and Marc Eisenstadt are gratefully acknowledged.

4.2.2 ITS'08 demonstrations

Demonstration of Agora UCS Ubiquitous Collaborative Space

Pascal Dugénie

LIRMM, Université Montpellier 2 & CNRS,
161 rue Ada,
Montpellier, France

Abstract. This demonstration consists to show some basic operations of the first AGORA UCS platform. This prototype has been developped on a new architecture designed for distributed learning with access to very large resource capacity and complex equipements. The two first nodes of the AGORA UCS infrastructure have been deployed in Montpellier (France) and Pisa (Italy). The access to this infrastructure can be done very easily from anywhere in the world. The aim of this demonstration is to show the large panel of learning context that can be covered and the extreme simplicity for any user who is not familiar with information technologies.

1 Agora UCS overview

The architecture of AGORA UCS is detailled in [1]. The originality of AGORA UCS is that it is designed for distributed learning as a side effect of communication and collaboration. There are no predefine learning goals but learning occurs quickly since the mastering of the environment is really simple. In order to achieve this, AGORA UCS architecture is based on three principles:

- **Ubiquity**. Community members are distributed in time and space. The access to the community resources is independant from their termial equipment. A thin client avoid any complex manipulation for the user.
- **Immanence**. The full internal control of the goals and the destiny of a community is required to allow any kind of learning activity. Mechanisms such as co-optation and rights delegation is permitted in order to achive the immanence principle.
- **Multi-modal communication**. The interactions between the members of the community are reinforced to give a better awarness of the learning activities. Typically these are synchronous services such as shared visualisation and audio-video-conferencing.

2 Description of the demonstration

This demonstration is based on experiments realised during the past two years with various communities of users. Here we consider a community of micro-

4.2.3 ECAP'08

Ubiquitous Collaborative Spaces: a step towards collective intelligence

Pascal Dugénie

Université de Montpellier & CNRS
LIRMM, Laboratoire d'Informatique de Robotique
et de Microélectronique de Montpellier,
161 rue Ada, Montpellier, France
Pascal.Dugenie@lirmm.fr

Abstract. This article argue our motivations to adopt some principles from the complex systems theory to study the behaviour of a collaboration process, then summarises design aspects of the concept called Ubiquitous Collaborative Spaces (UCS) to catalyse the collective intelligence.

1 The problem and its cause

Despite of the omnipresence of Information Technologies (IT) in our every day life, the collective intelligence as depicted in Pierre Levy's philosophy [?], still remains in the domain of utopia. An apparent reason behind this observation is the strong dependance of the design choices upon the users intentions. In fact, the final user is seldom actively involved during the system construction phase. Instead, in the classical model shown on the top of figure **??**, the user may interact with a collaborative system kernel via a human interface but has merely no possibilities to feed actions back to the the system structure.

Although IT have undoubtly attained today a suitable level of performance, the cybernetic idea of collective intelligence has not yet gained enough maturity in the mind of the system designers. A possible cause of this problem is the high complexity of designing systems with dynamic and self-adaptive behaviour. In one hand, how to design a flexible structure with no strict relations between elements of the system? In another hand, how to determine the finality of a collaborative system, since the intentions may change significantly during the collaboration process? These questions raise numerous challenges that we have considered in our conceptual approach.

2 A proposition to resolve the problem

As shown on the bottom part of the figure **??**, we have initially adopted a Service-Oriented Architecture (SOA) model to bootstrap the collaboration where both system designers and users take an active part in the system construction [?]. They may interact as agents of a global system by providing services to a common

4.2.4 SASO'08

Immanence in Grid

Pascal Dugenie

CNRS, Université de Montpellier
LIRMM, Laboratoire d'Informatique de Robotique
et de Microélectronique de Montpellier
161 rue Ada, Montpellier, France

Abstract. This position paper proposes to bring together the conception of GRID with the notion of immanence. On one hand, GRID is known to be an extraordinary infrastructure for coordinating distributed computing resources and Virtual Organisations (VOs). On one other hand, immanence is a principle that emerge from the internal behaviour of complex systems such as social organisations. Although several existing VO models specify how to manage resource, security policies and communities of users, none of them has considered to tackle the internal self-organisation aspect of the overall system.
We propose AGORA, a VO model integrated in an experimental collaborative environment platform. Its architecture adopts a novel design approach for VO management, modelled as a dynamic system in which the result of agent interactions are fed back into the system structure. Since the global behaviour of such a system emerges from its internal organisation, we place the immanence as the central AGORA principle for self-organising GRID VOs.

1 Introduction

1.1 The principle of immanence

Immanence usually refers to philosophical and metaphysical concepts. However, this principle expresses, in a wider sense, the idea of a strong interdependance between the *organisation* and the *activity* of a complex system (see 1.1). An immanent system is a system that constantly re-constructs its own structure throughout its internal activity. *The organisation is immanent to the activity.* By opposition, a system whose behaviour would be completely determined from the initial conditions with no feedback effect of its activity on its own structure is not an immanent sytem and has no chance to be self adaptive in case of changes of conditions of its environment. Examples of immanent systems are living systems in biology or social organisations. The principle of immanence has been introduced in informatics to describe the social impact derived from the introduction of the internet in the society. Furthermore, the expression *collective intelligence* has been adopted to describe the immanent system of knowledge structured around the Web [15]. The material (*i.e* the Web) is immanent to the immaterial (*i.e* the collective intelligence).

4.2.5 AWESOME'08

GRID-MAS integration and the principle of immanence

Pascal Dugenie

CNRS, Universite de Montpellier
LIRMM, Laboratoire d'Informatique de Robotique
et de Microelectronique de Montpellier
161 rue Ada, Montpellier, France

Abstract. This position paper proposes to bring together the integration of GRID and MAS with the notion of immanence. On one hand, GRID is known to be an extraordinary infrastructure for coordinating distributed computing resources and Virtual Organisations (VOs). On one other hand MAS interest focusses on complex behaviour of systems of agents. Immanence is a principle that emerges from the internal behaviour of complex systems such as social organisations. Although several existing VO models specify how to manage resource, security policies and communities of users, none of them has considered to tackle the internal self-organisation aspect of the overall system.

We propose AGORA, a VO model integrated in an experimental collaborative environment platform. Its architecture adopts a novel design approach, modelled as a dynamic system in which the result of agent interactions are fed back into the system structure. Since the global behaviour of such a system emerges from its internal organisation, we place the immanence principle as the main AGORA contribution for self-organising GRID VOs.

1 Introduction

1.1 The principle of immanence

Immanence usually refers to philosophical and metaphysical concepts. However, this principle expresses, in a wider sense, the idea of a strong interdependance between the *organisation* and the *activity* of a complex system (see 1.1). An immanent system is a system that constantly re-constructs its own structure throughout its internal activity. *The organisation is immanent to the activity.* By opposition, a system whose behaviour would be completely determined from the initial conditions with no feedback effect of its activity on its own structure is not an immanent sytem and has no chance to be self adaptive in case of changes of conditions of its environment. Examples of immanent systems are living systems in biology or social organisations. The principle of immanence has been introduced in informatics to describe the social impact derived from the introduction of the internet in the society. Furthermore, the expression *collective*

4.2.6 HPDC'08

Agora UCS: a Grid service for deploying Ubiquitous Collaborative Spaces

Pascal Dugénie and Stefano A. Cerri

Université de Montpellier & CNRS
LIRMM, Laboratoire d'Informatique de Robotique
et de Microélectronique de Montpellier,
161 rue Ada, Montpellier, France

Abstract. This paper proposes the architecture of a Grid service in order to deploy Ubiquitous Collaborative Spaces (UCS). This study originates from requiements in the area of Computer Supported Cooperative Work (CSCW). We realise that UCS is a suitable mean to enhance CSCW as long as a number of conditions are met. Current approaches endeavour to develop ah-hoc environments that corresponds to a given need but miss one or more requirements to become deployable as a generic service. The Agora UCS architecture model is characterised by a multimodal user interface for communication, and also more novative characteristics like ubiquity and immanence. Further recommendations such as security, high availability and QoS balancing has allowed us to outline the adequate infrastructure to operate AGORA. In the state of the art we analysed the potential of different service-oriented architectures to meet these recommendations. Initially, we realised that the Web lacks state management in services which does not allow to deploy ubiquitous services. Then, we clearly found a suitable candidate with the Grid architecture in terms of coordination of distributed resources and in terms of security. However, there is no satisfactory solution for the management of virtual organisations. The Grid community push models like Community Authorisation Service (CAS) or Virtual Organisation Management Service (VOMS) but both lack one of our essential characteristics, the immanence that we promote to ensure self-organisation of the communities. Thus, we have deeply explored this question and proposed an original solution for AGORA inspired from the Multi-Agent Systems (MAS). Furthermore, a worlwide experimentation plan with seventy users has shown the potential of the AGORA model. In a comparison with Access Grid, a widely used collaboration and visualisation system on Grid, AGORA present many advantages for the user as it is more easily accessible in a few mouse clicks.

1 Motivations

Sharing resources for performing intensive computation is the Grid moto where most research activity of this domain is focused. However, there is little number

Chapter 5

Conclusion and further work

There is a clear evidence that AG and Agora do not offer the same kind of service. AG is more focused on equiped meeting rooms whereas Agora UCS provides, at a lower cost, individual access to a common collaborative environment. Agora and AG have in common the fact that they improve the awareness capability, but Agora aims to enhance autonomy of the VO ahead of what AG can offer. Experimental results of Agora UCS platform indicates that a Grid infrastructure may open many significant perspectives at the condition that the effort is focused on the quality of service provision (i.e pertinence of sites location to deploy new resources) rather than a vertical integration of software layers (e.g the WSRF framework). For this reason, maintaining alive both Agora UCS nodes at LIRMM and ISTI will allow to continue these experiments and improve the collaborative environment service based validating mechanisms required by a distributed collaborative service.

This work, at a very early stage, has already contributed to new ways to approach complex system design where the self-organisation criteria is critical. We are aware that a serious validation process is still necessary in order to demonstrate that the *organisation* and the *activity* are completely interleaved and fully construct each other. However, our current achievements allow to envisage such a process, what is a progress, we believe, beyond the current state of the art.

Bibliography

[1] AGtk Quick start user guide. Technical report, Los Alamos National Laboratory, 2007. 15

[2] Stefano A. Cerri. Shifting the focus from control to communication: the STReams OBjects Environments model of communicating agents. In J.A. Padget, editor, *Collaboration between Human and Artificial Societies, Coordination and Agent-Based Distributed Computing, Lecture Notes in Artificial Intelligence*, volume 1624, pages 74–101, Berlin, Germany, 1999. Springer-Verlag. 7

[3] P. Dourish and V. Bellotti. Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM conference on Computer-Supported Cooperative Work*, pages 107–114, New York, 1992. CM Press. 6

[4] Pascal Dugénie. *UCS, Ubiquitous Collaborative Spaces on an infrastructure of distributed resources*. PhD thesis, University of Montpellier, 2007. In French, to be downloaded at http://www.lirmm.fr/˜dugenie/these. 7, 11, 18

[5] Pascal Dugénie. Ubiquitous Collaborative Spaces: a step towards collective intelligence. In *European Conference on Computing and Philosophy, ECAP'08*, Montpellier, June 2008. 7, 22

[6] Pascal Dugénie and Stefano A. Cerri. Agora, a model to exploit distributed resources for deploying Ubiquitous Collaborative Spaces. In *High Performance Distributed Computing*, 2008. submitted article currently in review process. 22

[7] Pascal Dugénie and Stefano A. Cerri. Immanence in Grid. In *Self-adaptive and Self-organizing Systems, SASO'08*, Venise, 2008. soumis le 19 mai 2008, en attente d'acceptation. 22

[8] Pascal Dugénie, Stefano A. Cerri, Philippe Lemoisson, and Abdelkader Gouaich. Agora UCS, Ubiquitous Collaborative Space. In *Intelligent*

Tutoring Systems, ITS'08, volume 5091 of *Lecture Notes in Computer Science*, pages 696–698, Heidelberg, 2008. Springer Berlin. 7, 22

[9] Pascal Dugénie, Clément Jonquet, and Stefano A. Cerri. The Principle of Immanence in GRID-Multiagent Integrated Systems. In *AWESOME*, 2008. in review process. 22

[10] Pascal Dugénie, Philippe Lemoisson, Clément Jonquet, Monica Crubézy, and Claude Laurenço. The Grid Shared Desktop: a bootstrapping environment for collaboration. In *Advanced Technology for Learning (ATL), Special issue on Collaborative Learning*, volume 3, pages 241–249, 2006. 7

[11] Frédéric Duvert, Clément Jonquet, Pascal Dugénie, and Stefano A. Cerri. Agent-Grid Integration Ontology. In R. Meersman, Z. Tari, and P. Herrero, editors, *International Workshop on Agents, Web Services and Ontologies Merging, AWeSOMe'06*, volume 4277 of *Lecture Notes in Computer Science*, pages 136–146, Montpellier, France, November 2006. Springer-Verlag. 10

[12] Jacques Ferber, Olivier Gutknecht, and Fabien Michel. From agents to organizations: an organizational view of multi-agent systems. In P. Giorgini, J. P. Müller, and J. Odell, editors, *4th International Workshop on Agent-Oriented Software Engineering, AOSE'03*, volume 2935 of *Lecture Notes in Computer Science*, pages 214–230, Melbourne, Australia, July 2003. Springer-Verlag. 10

[13] Ian Foster. Globus Toolkit Version 4: Software for service-oriented systems. *Journal of Computer Science and Technology*, pages 513–520, 2006. 14

[14] Ian Foster, Nicholas R. Jennings, and Carl Kesselman. Brain meets brawn: why Grid and agents need each other. In *3rd International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS'04*, volume 1, pages 8–15, New York, NY, USA, July 2004. 10

[15] Ian Foster, Carl Kesselman, Jeff Nick, and Steve Tuecke. The physiology of the Grid: an Open Grid Services Architecture for distributed systems integration. In *Open Grid Service Infrastructure WG, Global Grid Forum*. The Globus Alliance, June 2002. 10

[16] Ian Foster, Carl Kesselman, and Steve Tuecke. The anatomy of the Grid: enabling scalable virtual organizations. *Supercomputer Applications*, 15(3):200–222, 2001. 8

[17] ITU-T. H.263, infrastructure of audiovisual services, video coding for low bit rate communication. Technical report, International Telecommunication Union, 2005. 14

[18] Clément Jonquet and Stefano A. Cerri. The STROBE model: Dynamic Service Generation on the GRID. *Applied Artificial Intelligence Journal*, 19:967–1013, 2005. 7

[19] Clément Jonquet, Pascal Dugénie, and Stefano A. Cerri. Agent-Grid Integration Language. *International Journal on Multi-Agent and Grid Systems*, 4(2):167–211, 2008. 10

[20] Clément Jonquet, Pascal Dugénie, and Stefano A. Cerri. Service-Based Integration of Grid and Multi-Agent Systems Models. In R. Kowalczyk, M.N. Huhns, M. Klusch, Z. Maamar, and Q.B. Vo, editors, *International Workshop on Service-Oriented Computing: Agents, Semantics, and Engineering, SOCASE08*, volume 5006 of *Lecture Notes in Computer Science*, pages 56–68, Estoril, Portugal, May 2008. Springer-Verlag. 10

[21] Domenico Laforenza. Escogitare, Un'infrastruttura di E-SCience per gli Istituti di Ricerca del settOre aGro-alImenTARE. Technical report, CNR, Consiglio Nazionale delle Ricerche, 2006. 16

[22] Pierre Lévy. *L'intelligence collective. Pour une anthropologie du cyberspace*. La Découverte, Paris, 1994. 6

[23] Laura Pearlman, Von Welch, Ian Foster, Carl Kesselman, and Steven Tuecke. A Community Authorization Service for group collaboration. In *3rd International Workshop on Policies for Distributed Systems and Networks, POLICY'02*, pages 50–59, Monterey, CA, USA, June 2002. IEEE Computer Society. 8

[24] V. Ciaschini A. Frohner A. Gianoli K. Lorentey F. Spataro R. Alfieri, R. Cecchini. Voms: an authorization system for virtual organizations. In *Grid Computing*, volume 2970 of *Lecture Notes in Computer Science*, pages 33–40, Heidelberg, 2004. Springer Berlin. 8

[25] Cindy Sievers. Access Grid Venue Client 3.x User Manual. Technical report, Los Alamos National Laboratory, 2007. 14, 15

[26] Marc Weiser and John Seely Brown. Designing calm technology. *PowerGrid Journal*, July 1996. 5