The Principle of Immanence in GRID-Multiagent Integrated Systems

Pascal Dugénie, Clement Jonquet, Stefano A. Cerri

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* *intelligence* has been adopted to describe the immanent system of knowledge structured around the Web [17]. 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.

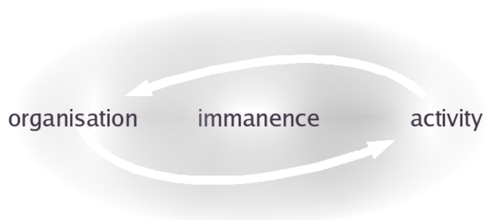

Fig. 1. Principle of immanence

1.2 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 [1,14]. 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 [5,2,3,4]. 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 A brief state-of-the art

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[1]. The aspect of *VO management* is studied through various conceptual models of organisation.

2.1 Grid and MAS convergence

The convergence of Grid and MAS research activities brings up new perspectives towards a immanent system. Firstly, Grid SOA[2] 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* [11]. 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 [8].

Although, the integration of Grid SOA and MAS concepts has been proposed to enhance the potential of organising distributed resources and agents [10], 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 [6,16,15]. 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.

2.2 Virtual Collaborative Environments

A Virtual Collaborative Environment (VCE) is a concept mainly studied in the domain of CSCW. 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. The multiplicity of modalities of communication (*i.e* multimodal communication) lays on the principle that increasing the *awareness* improves the efficiency of the collaboration process.

[1] CSCW: Computer Suported Collaborative Work

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

2.3 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. Furthermore, this access benefits from the principle of *ubiquity* since Grid resources are able to maintain the state of the communications independently from the location of the terminal elements.

Access Grid (AG) is the most world-wide deployed Grid VCE. AG operates on *Globus* [9], the most popular Grid middleware. The topology of the AG infrastructure consists of two kinds of AG nodes [19]: the *venue clients* and the *venue servers*. AG venue clients can meet in a venue server to set up a meeting. AG uses the H.263 protocol [13] for audio and video encoding 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.

2.4 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 [12]. 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 [18]. The functionalities for VO membership and rights management are centralised in a LDAP[3] 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) [7] 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.

2.5 Discussion

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

[3] Lightweight Directory Access Protocol

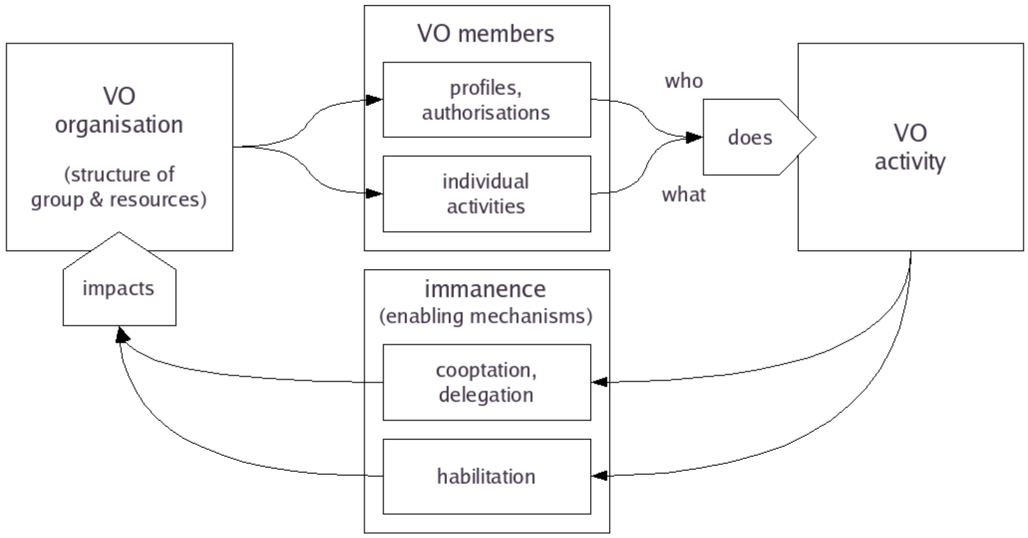

Fig. 2. A typical VO management model

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

3 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[4] 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 [2].

3.1 Conceptual model

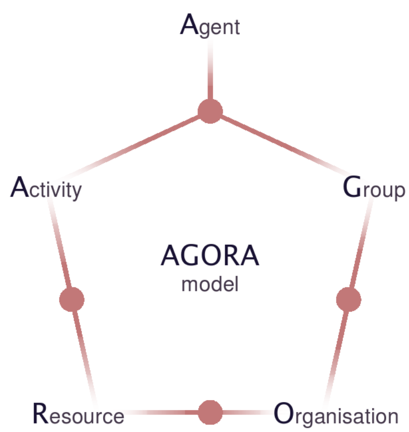

Fig. 3. AGORA conceptual model

The conceptual model of AGORA UCS presented on figure 3.1 consists of a set of **five concepts** and **four relations**:

Agent . This concept constitutes the main driving element that can change the state of the overall system. It's type may be a artificial (a process) or a human.

Group . This concept contains a number of agents who are considered as members of this group. An agent may be a member of several groups and exerce a different activity according to the group. A group can be seen as the context of a that activity.

Organisation . This concept is formed with one given group and one given set of resource.

---

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

It is a bijective so that a given group is associated with a resource and one, and vice versa.

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 *container services*[5].

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.

A ternary relation between the three concepts, *agent*, *group* and *activity*, enables to resolve the limitation for self-organisation of existing VO management models. This relation expresses that an agent may become member of one or several groups and exerce different activities in the context of one of these groups. Another important aspect of this model are the two bijective relations: one between a given *organisation* and a group of agents (*i.e* a *community*) and one between this *organisation* and a given set of resource (*i.e* a *service container*). A *service container* ensure the provision of resource to the *community*.

3.2 Persistent Core Services

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

Figure 3.2 is a representation based on the AGIL formalism, that shows a service coontainer including six PCS:

1. $\boxed{\textbf{A}}$ **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. $\boxed{\textbf{M}}$ **embers**: A VO is composed of members. This PCS manages the description of members of a VO, adding or removing members.
3. $\boxed{\textbf{G}}$ **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. $\boxed{\textbf{H}}$ **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. $\boxed{\textbf{E}}$ **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. $\boxed{\textbf{N}}$ **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.

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

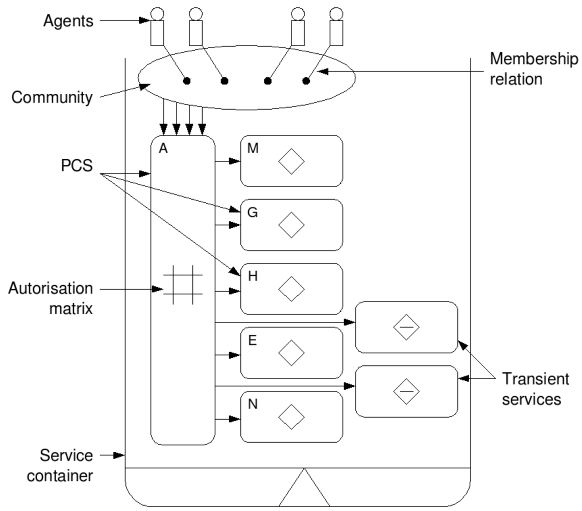

Fig. 4. The six PCS

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

3.3 Experimentations

Extensive experiments have 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.

For instance a scenario called EnCOrE[6] 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

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

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 Conclusion

Can we say that the principle of immanence has been achieved? We could answer that it partially occured in anecdoctical situations during our experiments. But the most important is that the strenght of the AGORA UCS conceptual model, the experiments on the platform and all the promises of the GRID infrastructure open many significant perspectives. 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.

References

1. 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. 2
2. 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. 2, 6
3. Pascal Dugénie. Ubiquitous Collaborative Spaces: a step towards collective intelligence. In *European Conference on Computing and Philosophy, ECAP'08*, Montpellier, June 2008. 2
4. 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. 2
5. Pascal Dugénie, Philippe Lemoisson, Clément Jonquet, Monica Crubézy, and Claude Laurençо. 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. 2
6. 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. 3

7. R. Alfieri et al. 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. 4
8. 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. 3
9. Ian Foster. Globus Toolkit Version 4: Software for service-oriented systems. *Journal of Computer Science and Technology*, pages 513–520, 2006. 4
10. 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. 3
11. 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. 3
12. Ian Foster, Carl Kesselman, and Steve Tuecke. The anatomy of the Grid: enabling scalable virtual organizations. *Supercomputer Applications*, 15(3):200–222, 2001. 4
13. ITU-T. H.263, infrastructure of audiovisual services, video coding for low bit rate communication. Technical report, International Telecommunication Union, 2005. 4
14. Clément Jonquet and Stefano A. Cerri. The STROBE model: Dynamic Service Generation on the GRID. *Applied Artificial Intelligence Journal*, 19:967–1013, 2005. 2
15. 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. 3
16. 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. 3
17. Pierre Lévy. *L'intelligence collective. Pour une anthropologie du cyberspace*. La Découverte, Paris, 1994. 2
18. 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. 4
19. Cindy Sievers. Access Grid Venue Client 3.x User Manual. Technical report, Los Alamos National Laboratory, 2007. 4