The Principle of Immanence in GRID-Multiagent Integrated Systems

Pascal Dugenie[1], Clement Jonquet[1,2], and Stefano A. Cerri[1]

[1] Laboratory of Informatics, Robotics, and Microelectronics of Montpellier (LIRMM)
National Center of Scientific Research (CNRS) & University Montpellier 2
161 Rue Ada, 34392 Montpellier, France
{dugenie,cerri}@lirmm.fr
[2] Stanford Center for Biomedical Informatics Research (BMIR)
Stanford University School of Medicine
Medical School Office Building, Room X-215
251 Campus Drive, Stanford, CA 94305-5479 USA
jonquet@stanford.edu

Abstract. Immanence reflects the principle of emergence of something new from inside a complex system (by opposition to transcendence). For example, immanence occurs when social organization emerges from the internal behaviour of a complex system. In this position paper, we defend the vision that the integration of the GRID and Multi-Agent System (MAS) models enables immanence to occur in the corresponding integrated systems and allows self-organization. On one hand, GRID is known to be an extraordinary infrastructure for coordinating distributed computing resources and Virtual Organizations (VOs). On the other hand MAS interest focusses on complex behaviour of systems of agents. Although several existing VO models specify how to manage resource, services, security policies and communities of users, none of them has considered to tackle the internal self-organization aspect of the overall complex system. We briefly present AGORA, a virtual organization model integrated in an experimental collaborative environment platform. AGORA's 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.

1 Introduction

Immanence usually refers to philosophical and metaphysical concepts. However, immanence expresses, in a wider sense, the idea of a strong interdependence between the *organization* and the *activity* of a complex system. An system is immanent if it constantly re-constructs its own structural organization 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 system 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 organizations. 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 [1]: The material (*i.e.*, the Web) is immanent to the immaterial (*i.e.*, the collective intelligence).

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 feasibility. Immanence is becoming highly critical in the analysis of complex problems and therefore, there are several reasons for considering the potential of immanence in collaborative environments based on the integration of GRID and Multi-Agent System (MAS):

One reason is the possibility offered by the GRID infrastructure to deploy autonomous services [2].[3] These services can be instantiated in specific service container with their dedicated resources, 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 MAS. In this approach, interactions between agents are contextualized within a global collaborative process. The notion of agent is extended to cover artificial processes as well as human ones. Agents interact within a collaborative environment by providing or using services. Moreover, they can behave intelligently with those 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 [3]. 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-organization.

In this paper, we defend the vision that the integration of the GRID and Multi-Agent System (MAS) models enables immanence to occur in the corresponding integrated systems and allows self-organization. In such systems, immanence constitutes the *living link* between the organization (*i.e.*, the static model) and the activity (*i.e.*, the dynamic model), and both models act upon each other. The organization enables to generate the activity whereas the activity constantly seeks to improve the organization. To illustrate our vision, we briefly present the AGORA *ubiquitous collaborative space*, a virtual collaborative environment platform that benefits from an original immanent Virtual Organization (VO) management system thanks to its GRID-MAS based underlying model.

2 A brief state-of-the art

Most of the GRID research activity eludes the question of immanence because two complementary aspects are usually treated separately while they should be treated together.

[3] An service-oriented architecture has been adopted in the Open Grid Service Architecture that has become the reference model for GRID systems.

The *collaboration* aspect is treated in general within the domain of Computer Supported Collaborative/Cooperative Work. The *VO management* aspect is studied through various conceptual models of organization.

2.1 Virtual Collaborative Environments

Virtual Collaborative Environments (VCEs) are mainly studied in the domain of Computer Supported Collaborative/Cooperative Work. 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 visualization, instant messaging, 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. The advantage of a GRID infrastructure for a VCE is to allow 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 (www.accessgrid.org)is the most world-wide deployed GRID VCE. Access Grid operates on *Globus* [4], the most popular GRID middleware. The topology of the Access Grid infrastructure consists of two kinds of nodes [5]: the venue clients and the venue servers. Access Grid venue clients can meet in a venue server to set up a meeting. Access Grid uses the H.263 protocol [6] 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 Access Grid allows to add new features such as application sharing (shared desktop, presentation, etc.) and data sharing. Access Grid focusses on the principles of *awareness* and *ubiquity*. However, Access Grid 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.

2.2 Virtual Organization 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 [7]. 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. For examples:

Community Authorization Service (CAS) has been specifically designed to facilitate the management of large VO [8]. The functionalities for VO membership and rights management are centralized in a Lightweight Directory Access Protocol (LDAP) directory. Since, the structure of the VO is strongly hierarchical, this is hardly possible to reorganize the initial tree once the services are deployed.

Virtual Organization Membership Service (VOMS) [9] is deployed in more recent GRID infrastructures such as in the EGEE project (http://eu-egee.org/). 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 children. 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.

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.

Figure 1 represents a typical VO management system. It illustrates why both CAS and VOMS fail in ensuring self-organization of VOs. On the left part, the VO management system determines the overall system organization. On the middle part, mechanisms for VO member management and, on the right part, the resulting activity is directly dependent on the initial organization. At this stage, there is no more possibility to re-introduce activities back to the system organization. The bottom part of the figure represents the mechanisms for enabling the principle of immanence. 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.

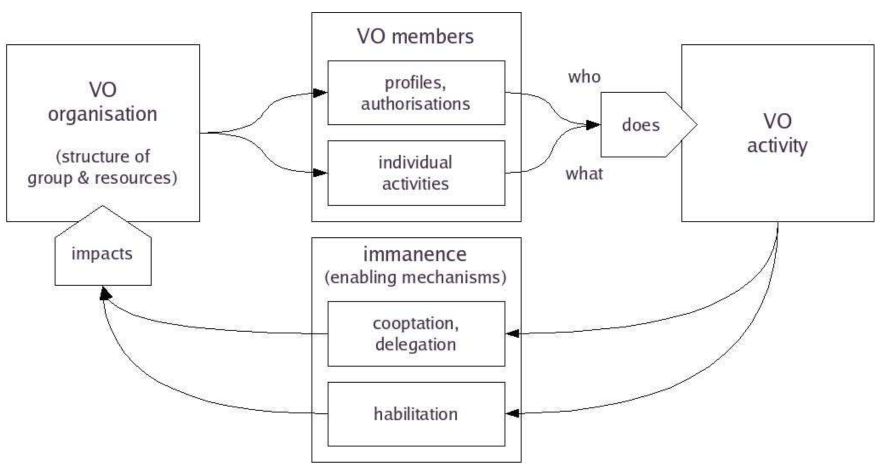

Fig. 1. A typical VO management model

2.3 GRID and MAS convergence

The GRID and MAS communities believe in the potential of GRID and MAS to enhance each other because these models have developed significant complementarities [10].

One of the crucial explorations concerns the substitution by an agent-oriented kernel of the current object-oriented kernel of services available in service-oriented architectures. The community agrees that such a change will really leverage service scenarios by providing new types of services [11]. This key concept of *service* is clearly at the intersection of the GRID and MAS domains and motivate their integration [12]. In [13], we introduce the *Agent-Grid Integration Language* (AGIL) as a GRID-MAS integrated systems description language which rigorously formalizes both key GRID and MAS concepts, their relations and the rules of their integration with graphical representations and a set-theory formalization. AGIL concepts are used in section 3 to illustrate the Agora ubiquitous collaborative space architecture (figure 3). AGIL represents an integration model in which we consider agents exchanging services through VOs they are members of: both the service user and the service provider are considered to be agents. They may decide to make available one of their capabilities in a certain VO but not in another. The VO's service container is then used as a service publication/retrieval platform. A service is executed by an agent with resources allocated by the service container. We sum-up here AGIL's two main underlying ideas:

- The representation of agent capabilities as Grid services in service containers, i.e., viewing Grid service as an 'interface' of an agent capability;
- The assimilation of the service instantiation mechanism – fundamental in GRID as it allows Grid services to be stateful and dynamic – with the mechanism to create dedicated conversation contexts fundamental in MASs.

Instead of being centred on how each entities of the system should behave, both GRID and MAS have chosen an organizational perspective in their descriptions. In organization centred MAS [14, 15], the concepts of organizations, groups, roles, or communities play an important role. In particular, Ferber et al. [15] presents the main drawbacks of agent-centred MAS and proposes a very concise and minimal organization centred model called Agent-Group-Role (AGR) from which AGIL is inspired as summarized in table 1. GRID-MAS integrated system benefit from both GRID and MAS organizational structure formalisms. Therefore, the convergence of GRID and MAS research activities brings up new perspectives towards a immanent system.

3 The AGORA Ubiquitous Collaborative Space

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 implemented and deployed 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 [16] (cf. section 3.3).

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

Table 1. AGIL's organizational-structure analogies between GRID and MAS

MAS	GRID
Agent	Grid user
An agent is an active, communicating entity playing roles and delegating tasks within groups. An agent may be a member of several groups, and may hold multiple roles (in different groups).	A Grid user is an active, communicating entity providing and using services within a VO. A Grid user may be a member of multiple VOs, and may provide or use several services (in different VOs).
Group	VO
A group is a set of (one or several) agents sharing some common characteristics and/or goals. A group is used as a context for a pattern of activities and for partitioning organizations. Two agents may communicate only if they are members of the same group. An agent transforms some of its capabilities into roles (abstract representation of functional positions) when it integrates into a group.	A VO is a set of (one or several) Grid users sharing some common objectives. A VO and the associated service container is used as a context for executing services and for partitioning the entire community of Grid users. Two Grid users may exchange (provide/use) services only if they are members of the same VO. A Grid user publishes some of its capabilities into services when it integrates into a VO.
Role	Service
The role is the abstract representation of a functional position of an agent in a group. A role is defined within a group structure. An agent may play several roles in several groups. Roles are local to groups, and a role must be requested by an agent. A role may be played by several agents.	The service is the abstract representation of a functional position of a Grid user in a VO. A service is accessible via the CAS service. A Grid user may provide or use several services in several VOs. Services are local to VOs (situated in the associate container), and a Grid user must be allowed to provide or use services in a VO. A service may be provided by several Grid users.

AGORA's ubiquity principle enables access to the VCE from anywhere at anytime. Although this principle has been envisaged several years ago [17], the concrete deployment of operational solutions has been feasible only recently, by means of pervasive technologies such as GRID. AGORA's awareness principle enable enhancing presence ([18]) by means of multimodal communication such as asynchronous file repositories or messaging as well as synchronous services such as audio-video communication, text chat and shared desktop. An extensive description of these two first characteristics is out of the scope of this article. We rather focus here on the self-organization capability of the VO that is described by the third characteristic of *immanence*, which is the characteristic which really distinguishes the AGORA UCS from related works such as Access Grid.

3.1 Conceptual model

The conceptual model of AGORA UCS presented on figure 2 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 plays a different activity according to the group. A group can be seen as the context of a that activity.

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

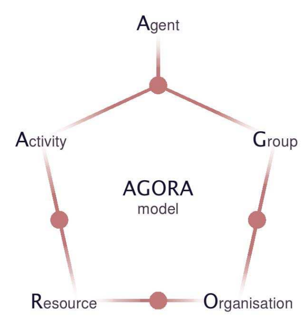

Fig. 2. AGORA UCS's conceptual model

A ternary relation between the three concepts, *agent*, *group* and *activity*, enables to resolve the limitation for self-organization of existing VO management models. This relation expresses that an agent may become member of one or several groups and play 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 *organization* and a group of agents (*i.e* a *community*) and one between this organization 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 persistent core services necessary for bootstrapping and maintaining a collaborative environment. Since one service container is associated to a VO, there are as many sets of persistent core services as VO. Figure

[5] GRID service containers can be defined as the reification of a portion of computing resources that have been previously virtualized through a GRID middleware.

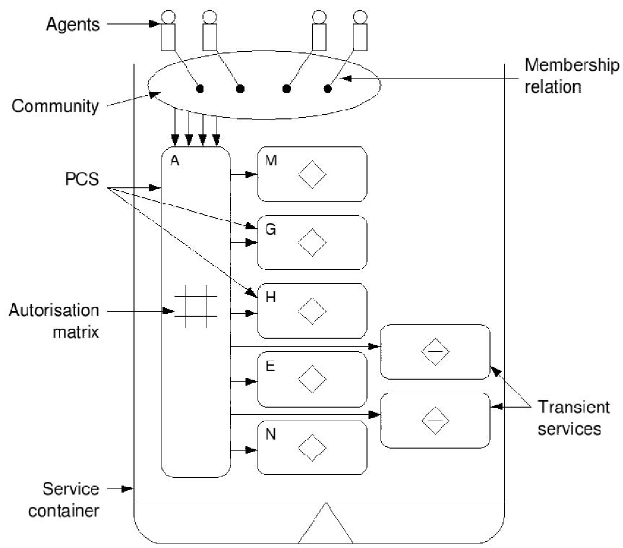

Fig. 3. AGORA UCS's persistent core services

3 is a representation that uses the AGIL [13] to shows an AGORA service container including the six persistent core services:

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 persistent core services.
2. **M embers**: A VO is composed of members. This service manages the description of members of a VO, adding or removing members.
3. **G roup**: A VO is characterized by its properties (identifier, description, etc.). Also, the creation of a new VO is always performed in the context of another VO. Therefore this service is in charge of both intra VO operations as well as extra VO operations.
4. **H istory**: All the data belonging to a VO must be stored, maintained and also indexed. This service is in charge of keeping track of changes, logs of events and also of recording collaboration sessions.
5. **E nvironment**: A VO may personalize its own environment. This environment operates in a service container. This service is in charge of adding or removing services (excluding the persistent core services).
6. **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.

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 visualization enabled a fast transfer of knowledge 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, chemistry scientists, 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 foreseen in advance, the flexibility of the AGORA UCS is essential to enable a community to freely organize 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 Conclusion

Can we say that the principle of immanence has been achieved? We could answer that it partially occurred in anecdoctical situations during our experiments. But the most important is that the strength 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-organization criteria is critical.

We are aware that a serious validation process is still necessary in order to demonstrate that the *organization* and the *activity* are completely interleaved and fully constructed each other. The convergence of GRID and Multi-Agents systems had revealed some interesting features to accomplish this view.

References

1. Lévy, P.: Collective Intelligence: Mankind's Emerging World in Cyberspace. Basic Books (1999)
2. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: 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)

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

3. Dugénie, P., Cerri, S.A., Lemoisson, P., Gouaich, A.: Agora UCS, Ubiquitous Collaborative Space. In: Intelligent Tutoring Systems, ITS'08. Volume 5091 of Lecture Notes in Computer Science., Heidelberg, Springer Berlin (2008) 696–698
4. Foster, I.: Globus Toolkit Version 4: Software for service-oriented systems. Journal of Computer Science and Technology (2006) 513–520
5. Sievers, C.: Access Grid Venue Client 3.x User Manual. Technical report, Los Alamos National Laboratory (2007)
6. ITU-T: H.263, infrastructure of audiovisual services, video coding for low bit rate communication. Technical report, International Telecommunication Union (2005)
7. Foster, I., Kesselman, C., Tuecke, S.: The anatomy of the Grid: enabling scalable virtual organizations. Supercomputer Applications **15**(3) (2001) 200–222
8. Pearlman, L., Welch, V., Foster, I., Kesselman, C., Tuecke, S.: A Community Authorization Service for group collaboration. In: 3rd International Workshop on Policies for Distributed Systems and Networks, POLICY'02, Monterey, CA, USA, IEEE Computer Society (June 2002) 50–59
9. Alfieri, R., Cecchini, R., Ciaschini, V., dell Agnello, L., Frohner, A., Gianoli, A., Lõrentey, K., Spataro, F.: Voms: an authorization system for virtual organizations. In: Grid Computing. Volume 2970 of Lecture Notes in Computer Science., Heidelberg, Springer Berlin (2004) 33–40
10. Foster, I., Jennings, N.R., Kesselman, C.: 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., New York, NY, USA (July 2004) 8–15
11. Huhns, M.N., Singh, M.P., Burstein, M., Decker, K., Durfee, E., Finin, T., Gasser, L., Goradia, H., Jennings, N.R., Lakkaraju, K., Nakashima, H., Parunak, V., Rosenschein, J.S., Ruvinsky, A., Sukthankar, G., Swarup, S., Sycara, K., Tambe, M., Wagner, T., Zavala, L.: Research directions for service-oriented multiagent systems. Internet Computing **9**(6) (November-December 2005) 65–70
12. Jonquet, C., Dugenie, P., Cerri, S.A.: Service-Based Integration of Grid and Multi-Agent Systems Models. In Kowalczyk, R., Huhns, M., Klusch, M., Maamar, Z., Vo, Q., eds.: International Workshop on Service-Oriented Computing: Agents, Semantics, and Engineering, SOCASE'08. Volume 5006 of Lecture Notes in Computer Science., Estoril, Portugal, Springer-Verlag (May 2008) 56–68
13. Jonquet, C., Dugenie, P., Cerri, S.A.: Agent-Grid Integration Language. Multiagent and Grid Systems **4**(2) (2008) 167–211
14. Wooldridge, M., Jennings, N.R., Kinny, D.: The Gaia methodology for agent-oriented analysis and design. Autonomous Agents and Multi-Agent Systems **3**(3) (September 2000) 285–312
15. Ferber, J., Gutknecht, O., Michel, F.: From agents to organizations: an organizational view of multi-agent systems. In Giorgini, P., Müller, J.P., Odell, J., eds.: 4th International Workshop on Agent-Oriented Software Engineering, AOSE'03. Volume 2935 of Lecture Notes in Computer Science., Melbourne, Australia, Springer-Verlag (July 2003) 214–230
16. Dugénie, P.: 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.
17. Weiser, M., Brown, J.S.: Designing calm technology. PowerGrid Journal (July 1996)
18. Eisenstadt, M., Dzbor, M.: BuddySpace: Enhanced Presence management for collaborative learning, working, gaming and beyond. In: JabberConf Europe, Munich, Germany (June 2002)