Socially-aware lightweight coordination infrastructures

Marc Esteva, Juan A. Rodriguez-Aguilar, Josep Lluís Arcos, and Carles Sierra

Artificial Intelligence Research Institute (IIIA-CSIC)
Spanish National Research Council
08193 Bellaterra, Spain
{marc,jar,arcos,sierra}@iiia.csic.es

Abstract. Coordination infrastructures have played a central role in the engineering of multi-agent systems (MAS). Although MAS research has produced a number of notable coordination infrastructures of varying features, these have been mainly conceived to host software agents and to facilitate multi-agent programming. Thus, human agents have been mostly kept out of the picture, hence hindering the use of agent-oriented infrastructures for the coordination of hybrid multi-agent systems. Moreover, current MAS tools supporting coordination heavily rely on a *dedicated* infrastructure. In this work we touch upon these two issues. On the one hand, we analyse the kind of coordination support required by humans in a hybrid multi-agent system. On the other hand, we propose how to achieve coordination with little, lightweight infrastructure.

1 Introduction

Multiagent systems (MAS) are composed by a group of agents that interact within an environment to achieve their common or individual goals. Typically, the achievement of such goals require the effective coordination of participant behaviours. From an engineering point of view, building a MAS entails the problems of designing a distributed concurrent system plus additional difficulties due to the autonomy of their entities. Therefore, the development of infrastructures that provide support to agent coordination play a central role in the engineering of MAS. While, initially, each MAS was designed *ad hoc* and developed its own infrastructure from scratch [1], as the area has evolved, certain tasks have been abstracted and gradually provided by MAS infrastructure as domain independent functionalities or services. The services offered by infrastructures vary from providing communication and yellow pages services, as for instance in FIPA compliant platforms [2], to give support to the execution of agent organisations or institutions [3, 4]. In this later case, the infrastructure supports the execution of the system following the coordination model defined by the regulations imposed by the organisation or institution. Furthermore, having a explicit representation of the coordination model, which is the case in most of these approaches, facilitates its run-time adaptation to continue being effective under changing circumstances.

The purpose of this paper is two-fold: (i) to review the state of the art in coordination infrastructures for MAS, and (ii) to present what we identify as the main challenges to face for the next generation of coordination infrastructures. Regarding the later, we advocate supporting human participation as one of the main challenges. In other words, to support the execution of hybrid systems where participants may be human and software agents. Notice that this systems can be regarded as a kind of open systems, and thus, they can be engineered following a MAS approach. However, developed infrastructures mainly support the participation of software agents, and the human role is limited to acting behind the scenes by customising provided agent templates that participate in the system on humans' behalf. We regard this as a limitation to the spread of agent technologies, since in many applications users are reluctant to delegate their participation in the system to an agent. For instance, in e-commerce applications where the participation in the system may imply spending the user money. We advocate, that supporting the execution of this kind of hybrid systems would increase the use of MAS technologies. In order to support human participation appropriate tools and interfaces have to be developed.

Normally, human organisations offer services that help users to participate in the system and achieve their goals. For instance, these services may provide users with information about the organisation regulations, or the steps to follow to achieve a certain goal. We regard the definition and incorporation of this kind of services, that we call *assistance services*, into coordination infrastructures as another challenge for the area. Notice that this services can provide assistance to both human and software agents.

Regulated environments for multiagent systems usually rely on a heavy infrastructure that has to be up and running before the agents can actually interact. As the central aspect of this kind of approaches is the idea that participants must accept the norms, defined at design time, that regulate it. This must happen before any action is performed by an agent. Otherwise, the consequences of the actions might be unknown by agents, and therefore any subsequent sanction would be unfair. We argue that a peer to peer (P2P) approach can provide a more light-weight infrastructure for agent coordination. The essence of P2P computing is that peers when active can be called on duty to give to the other peers knowledge declared as available and to execute services or agents that are also made available to the rest. This view permits community creation, facilitates the sharing of resources, and when created appropriately (i.e. by giving incentives to their usage) permits an explosion in usage. A key component of this approach is a repository of declarative specification of coordination patterns. Hence, agents can reason about them and instantiate the ones needed to coordinate and to achieve their goals. We illustrate this approach by applying it to electronic institutions. The result is a lightweight infrastructure and a click-away from usage.

The rest of the paper is structured as follows. Next, in section 2 we review the state of the art in coordination infrastructures. Thereafter, in section 3 we focus on supporting human participation (3.1), and assistance services (3.2). In section 4 we present a proposal of a lightweight infrastructure based on a P2P approach, while in section 5 we conclude.

2 State of the art

As above mentioned as MAS research has evolved different coordination infrastructures providing more domain independent services have been developed. Hence, they can be reused in the the deployment of different applications reducing the development time and cost. We regard the FIPA specification for agent platforms as the first important improvement in order to provide coordination infrastructures for MAS [2]. In particular, FIPA proposed that any agent platform has to provide the following services: (i) an agent management service that manages agents executions and permits to find other agents (i.e. white pages service); (ii) and a directory service to register and discover agent services; and (iii) a reliable transport service among agents. Hence, FIPA compliant infrastructures, such as JADE [5], must provide these services to participating agents facilitating agent coordination by offering mechanisms to discover other agents and the services they offer, and a transport service transparent to agent physical locations

Organisational approaches provide a higher level abstraction to define the coordination model among agents. In general all of them include a social structure defining the roles agents may play, and a set of regulations or conventions that structure participants behaviours. Some organizational approaches use FIPA compliant platforms to support their execution, but they provide limited support to organisational concepts. However, specific infrastructures have been developed to support organisational models, as for instance AMELI [4] and S-Moise^{+} [3], In particular, AMELI supports the execution of any electronic institutions specified using the ISLANDER editor. An institution specification defines the social structure, the dialogical interactions (scenes), agents can engage on, and the norms that establish the consequences of their. In order to support the institution execution AMELI implements the necessary mechanisms to enforce the correct execution of the institution according to its specification. One of its main features is that it accepts the participation of agents in any language and internal architecture, S-Moise^{+} is an infrastructure to run organisations defined following the Moise model, which as an electronic institution it defines a social structure and the interactions agents may have. Moreover, it incorporates a functional specification that takes system tasks (derived from the organisational goals) and divides them into sub-tasks to be carried out by agents.

While previous infrastructures provide support to an execution of a specific organisational model, CArtAgO [6] is a framework to build infrastructures based on the Agents & Artifacts meta-model(A&A) [7]. This model proposes that apart of agents the environment is populated by a set of objects, called artifacts. Among other features artifacts can be used to coordinate agent activities. For instance, when an interaction protocol is instantiated an artifact can be created to coordinate the protocol execution. Rather than proposing an organisational model, CArtAgO can be used to build an infrastructure to support it. For instance, in OR4MAS [8] it is used to provide an infrastructure for the Moise model. That is, the system organisation is defined using the Moise model, and CArtAgO is used to build the system infrastructure.

Related to web services, the ALIVE framework was proposed to support the engineering of service-oriented systems [9]. ALIVE proposes three design and execution levels that incorporate coordination and organisation mechanisms as a way of facilitating dynamic capabilities to web services. Specifically, the organisational level is used to dynamically select, compose, and invoke services. Notice that the dynamic selection provides an adaptation of the coordination model depending on current situations.

This later issue, the adaptation of the coordination model to varying situations, is an important topic due to the dynamicity of MAS. Although we regard this as an open research issue for coordination infrastructures, there are some works that have addressed it. In particular, in S-Moise$^+$ there is a special role *Reorg* in charge of reorganising how the task are assigned to agents, while in [10] authors propose an extension to AMELI to endow it with self-adaptation capabilities. Moreover, there are other proposals to endow organisations with self-adaptation capabilities that can be incorporated into MAS infrastructures [11, 12].

3 Coordination support

3.1 Human computer interaction

We advocate that human incorporation is one of the main challenges for coordination infrastructures. In order to do so appropriate tools and interfaces have to be developed to address human requirements, which are different of software agent ones. Hence, they require different ways of participating. For instance, interacting in the system by exchanging messages in an agent communication language is appropriate for software agents but not for human users. Another key aspect is how to represent the relevant information to successfully participate in the system. This is information about the regulations of the coordination model, the system state and the actions performed by other participants. Notice that all this information determine the the valid actions a participant can do, and normally is used in its decision process.

Obviously, web pages or 2D interfaces can be used to facilitate human participation, but we advocate that more immersive environments as *3D virtual worlds* can do a better job in supporting human participation. *3D virtual worlds* is a technology that has emerged in nowadays computing with enormous strength. Humans are social and therefore the concept of virtual worlds is very appealing as they permit a much more immersive environment for their interactions. These are computational immersive environments that emulate real world using 3-dimensional visualisation. Humans participate in those environments represented as graphical embodied characters (avatars) and operate by using simple

Fig. 1. Snapshot of a Virtual Institution execution.

and intuitive control facilities. The immersive environment provides many possibilities for representing the system state, and the regulations defined in the coordination model. For instance, other participants are represented as avatars, and they appearance can be used to display the role they are playing. We argue that 3D Virtual Worlds technology can be successfully used for ÒopeningÓ Multiagent societies to humans

This idea was explored by Bogdanovich [13] who proposed the concept of virtual institution as a combination of electronic institutions and 3D virtual worlds. In this context, electronic institutions are used to define the regulations that structure participants interactions, while users participate in the system by controlling an avatar in an automatically generated representation of the institution in the virtual world. Ongoing activities (interaction protocols) are represented as 3D room in the generated virtual world. To support the execution of this kind of systems, he proposed a run time infrastructure where a middleware causally connects AMELI and the 3D virtual world. This causal connection is performed by transforming user actions in the virtual world to messages understood by AMELI and updating the visualisation whenever the electronic institution state changes. Hence, the run-time infrastructure maintains consistent the institution state and its representation in the virtual world. The idea was further explored in the context of the itchy feet project where a prototype for the tourism domain was developed [14]. Recently, Trescak et al [15] proposed an extension of the infrastructure that among other features supports the connection of several virtual worlds to the very same institution. Hence, the system allows users to participate from different virtual worlds. The result is an hybrid system which allows the participation of human and software agents.

Figure 1 displays a snapshot of an auction room of a virtual institution execution. We can regard that the room recreates auction rooms in real live, where buyers participants have to sit in one of the room chairs. Avatars representing software agents buyers are represented with blue skin, while the ones with green skin represent software agents controlling the auction execution. In particular, the one on the left is in charge of auctioning goods, while the one on the right is in charge of validating and announcing auction results. The appearance of avatars controlled by human users are decided by each participant. The panel on the wall is used to represent the information of the current auction round. Notice that the created representation permits the user to easily perceive who are the other participants in the auction room, the role they are playing along with information of the current state.

3.2 Assistance services

Participating in a MAS can be difficult due to the complexity of the system. On the one hand, agents must be aware of the coordination model that structure their interactions. Notice, that over time more complex coordination models are proposed, Furthermore, this coordination model can be adapted and change at run-time. On the other hand, the system state, the environment where agents are situated, and participants may dynamically change over time. Notice that both issues have to be taken into account by agents to achieve their goals, and in the decision making process. Hence, we propose that infrastructures should incorporate some services to help participants to successfully participate in the system and to achieve their goals. In other words, this services focus on assisting coordination. Notice that in general human organisations provide services and devote some resources to assist their users. Failing to provide adequate assistance to users may lead to the failure of the organisation. We regard assistance even more important for computational organisations. Notice that assistance services are useful for both software and human agents.

This services can vary from providing agents with the necessary information about the coordination model and system state, to proposing plans to achieve a provided goal. In particular in [16] authors identified the following services: (1) providing agents with information to participate in the MAS, (2) justifying action result, (3) giving advice to agents, and (4) estimating action consequences. The basic one is the information service that provides information about both the current coordination model and system state. Currently, there are some MAS approaches that already provide some of this information. For example, in S-Moise+, the special agent in charge of the organisation (OrgManager) informs participants when they acquire new obligations, Besides, in an Electronic Institution , the special agent in charge of an interaction activity (Scene Manager), informs participants when an agent joins/leaves an interaction protocol. The second one, the justification service, gives agents an explanation about the result of an action. For instance, it can explain why an action has not been allowed to an agent, or why he has acquired a new obligation. The advice services provides agents with a set of alternate plans (action sequences), to achieve their goals. FInally, the estimation service provides agents information about the consequences that executing an action would have. For instance, about the obligations that an agent would acquire, The service is called estimation because the action is not really performed, so its consequences if later on the agent decides to execute it may be different.

Among these services we regard the advice service as the more complex and interesting one. Notice that the infrastructure may have more information about the system state than an agent, so, it may provide better plans. Furthermore, it simplifies an agent development and reasoning process, as the agent can reason about provided plans to achieve its goals, and do not need to compute them itself. Information can be provided pro-actively by the infrastructure, as for instance, when the coordination model is adapted, or after a request by the agent. Furthermore, for some services participants can decide if they want to have the service active or not. In the case of human users the 3D immersive environment can be used to represent the information provided by each service.

4 Lightweight coordination

The essence of peer-to-peer (P2P) computing is that peers when active can be called on duty to give to the other peers knowledge declared as available and to execute services, or agents, that are also made available to the rest. This view permits community creation, facilitates the sharing of resources, and when created appropriately (i.e. by giving incentives to their usage) permits an explosion in usage. The purpose of this section is to introduce a P2P model for electronic institutions (EIs). As a result, we shall obtain a coordination infrastructure that is light and click-away from usage. Moreover, further interesting features (inherent to P2P systems) are inherited: high degree of decentralization, self-organisation, low barrier to deployment (compared to client-server systems), organic growth, resilience to faults and attacks, and abundance and diversity of resources [17]. Notice that although we focus our discussion on electronic institutions, these have been taken as a case study. Further research should aim at analysing the generality of our proposal.

4.1 A peer-to-peer model

The idea behind the architecture is already present in the concept of governor in electronic institutions. The interactions of each agent within an EI are mediated by a governor, which only accepts as valid the agent's actions that abide by the rules of the institution. This simple idea seems that can now become the base of a kernel for a P2P node in a network. The proposal is based on a P2P node that should be a downloadable component helping in: (i) *knowledge management*: sharing EI related materials; (ii) *search*: searching for components; and (iii) *distributed execution*: supporting the enactment of EI executions transparently to a human/agent user.

More concretely, the two sides of the node could be based on the following design principles:

– *Repository of EI specification components*. Each node may publish EI components: ontologies, scenes, norms, etc. The node could either publicise their own components or may make available components found elsewhere in the network. These components could be written in the current XML language used in ISLANDER with some extra annotations to help on provenance or in certificates. The node should incorporate a tool to permit the combination of specifications to constitute EIs. An institution can be seen then as a distributed performative structure where some sub-performative and/or scenes are located in different nodes. This opens the possibility of versioning, or local modifications that are immediately incorporated into those institutions that combine a particular bit being modified. For instance we could imagine that many institutions contain a sub-performative structure developed and maintained by Verisign in order to check for certificates. What institutions are made visible to the community must be a decision of the node manager.
– Repository of institutional agents. Agents playing the basic roles of 'Governor', 'Boss', 'Scene manager', and 'Transition manager' should be made available in many/most/all nodes in the network. These agents must be certified by their creators as they might be provide by different manufacturers with different levels of privacy guarantees, or different efficiency implementations. Creators of EIs might possible propose or give guidelines with respect to what are the preferred institutional agents to use.
– Repository of agents. Agents that can incarnate roles in EIs are also made available and shared through the node. Given a concrete EI agent creators will make publicly available agents that can play certain roles. These agents can be copied in different nodes according to users interest. Nodes should permit users to 'activate' agents by telling the node what agents playing what roles in which EI specification are allowed.
– Searching for components. One of the basic functionalities of the node should be about searching for different types of components: all those mentioned in the previous points plus trust information, ontology mappings, and running institutions.
– Execution support for a P2P infrastructure. A node might decide to enact an EI (e.g. I want to create an auction house following certain rules to sell second hand books). In order to do so, the P2P network should give the user the functionality to: (i) search for a node that incarnates the role of boss of a given specification, (ii) pass the control to the boss, and (iii) wait for the EI to get enacted and interact with it. To have this, the infrastructure (the boss) will need to recruit other institutional agents (scene managers and the like) through the network, publicise the execution to potential external agents (in our example, buyers) and monitor the execution of it. The state of the execution will therefore be distributed in the (hierarchical) organisation of

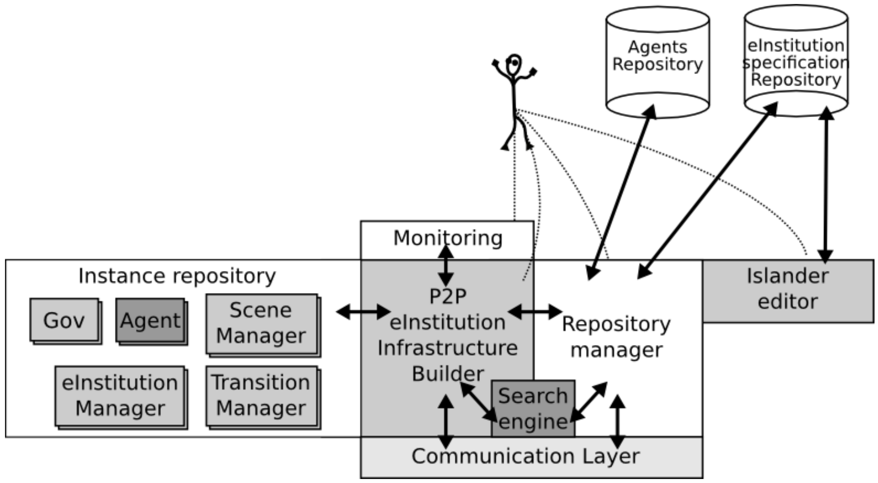

Fig. 2. Peer architecture and components

institutional agents that supports the execution. The failure of a node in the hierarchy would permit the P2P network to try and recover by appointing another agent to cover the role of the 'dead' agent. In some cases part of the state can perhaps be recovered if we keep backward paths in the organization, or if we allow for a more graph oriented organisation to increase robustness. The network should deal with the necessary routing to pass the messages among agents enacting the institution in a transparent way. Furthermore, it should be possible to put requirements for the execution of an EI (e.g. I only permit the participation of IPs that I know and trust).

4.2 A P2P architecture for electronic institutions

A P2P EI will be composed of peer nodes and a non-peer element, the Discovery Service.

P2P Electronic Institution Peer Figure 4.2 shows the architecture and components of a peer in a P2P EI, that we explain next. The communication between peers and the *Discovery Service*, will be through the *Communication Layer* component. This layer will abstract the communication process. For instance, we can think of using *JMS* (http://java.sun.com/products/jms/) through its *ActiveMQ* implementation (http://activemq.apache.org/). EI specifications are stored as XML files in the *EI specification Repository*.

Each peer also contain an *agents repository* storing implementations of Institutional and non-institutional agents. For each agent there must be a description about them: what EI is this agent coded for, the authors, a certificate, the agent version, etc. The agent implementation and this description can be bundled into a kind of *zip* file or *jar* file.

The *Repository manager* is the module in charge to publish into the *Discovery Service*: (i) the shared EI specifications available for download: (ii) the shared agent implementations available for download; (iii) the EI elements that the peer agrees to run for each EI specification: agents, governors, EI managers, scene managers, transition managers; and (iv) If the peer agrees to act as coordinator for an EI. Using the *Repository manager* the user can also search for resources published by other peers and select them to download. The needed search engine is provided by another peer component.

All instances of EI components that are currently participating at any EI are stored in the *Instance repository.* Inside the *Instance repository* we can find: agents, governors, EI managers, scene managers and transition managers. When an element at the *Instance repository* want to communicate with other elements, this communication is made through the *P2P EI Infrastructure Builder*. It knows whether the target is a local element located at the *Instance repository* and the message can be delivered locally, or is located at other peer and need to be sent through the *communication layer.*

Discovery Service (DHT) The Discovery Service provides a lookup service. It can be implemented by a DHT (Distributed hash table). This service is used by peers to inform about the resources that they share nd about the services that they can offer

4.3 A P2P Electronic Institution at work

Launching a P2P EI When a user wants to start a new EI, the XML EI specification and a set of restrictions are passed as parameters to the *P2P EI Infrastructure Builder*. This restrictions can include which peers are allowed to participate, what institutional elements are allowed to be instantiated by each peer, what agents implementations are allowed to participate, etc. Then, the user is asked if the *P2P EI* must be coordinated by his peer or if another coordinator is preferred. If another peer is preferred, a search process is launched. The *Search engine* is asked for available coordinators, and the returned list of available coordinators are asked if they agree to coordinate the P2P EI with that set of restrictions. If someone agrees, that peer will continue with the process of launching the P2P EI.

The peer playing as coordinator will publish itself at the *Discovery Service* as the P2P EI manager. Thereafter, by using the search engine, the coordinator asks to the *Discovery Service* about the available peers that agree to run each one of the needed elements. As a result, the search engine returns a list of the available peers, that is filtered by the coordinator to satisfy the restrictions. With the list of available peers, the recruiting process can be started. The recruiting process can be done in two ways: by hand (using the user interface) or automatically. Moreover, the coordinator can contain a reputation system to help to chose the more reliable peers. The recruiting consists in asking to the selected peers if they agree to participate into the P2P EI with the selected role (agent, governor, EI manager, scene manager or transition manager) and informing about the peer that is coordinating the P2P EI. If enough peers to start the execution of the EI accept, the process continues and the EI is launched. Otherwise, the process is aborted.

The first element to be instantiated is the *EI manager*. So the coordinator sends a message to the corresponding peer to create the *EI manager* and wait for confirmation. Each time that the *EI manager* needs to create an institutional agent, scene manager or transition manager, it will ask to the coordinator to create it, and will wait for confirmation.

Joining a running P2P EI A user can decide to join with an agent a currently running P2P EI. To perform this task, he asks the *Search Engine* about the currently running P2P EIs executing the desired EI. The *Search Engine* returns a list of coordinators that manage running instances of the desired EI. The next step is to ask the coordinator if it's possible to participate. If the peer can participate, the coordinator returns the address of the peer running the *EI Manager*, so the user can request that agent to enter the EI playing the desired role. If accepted the *EI Manager* replays with the address of the *Governor* assigned to the agent. Thereafter, all the interaction with the EI will be done through that *Governor*.

5 Conclusions

Coordination infrastructures play a central role for the engineering of multiagent systems. Over years researchers have developed infrastructures providing more domain independent services and capable of enacting more complex coordination models. In the first part of the paper we have revised the state of the art in this field.

Later on, we have shift our attention to the next challenges to be faced by the next generation of infrastructures. The first one is to support human participation. In other words, infrastructures should incorporate the necessary tools and interfaces to "open" MAS to humans and therefore, be socially-aware. We regard 3D Virtual Worlds as an appropriate technology for this task. Another important issue is the development of assistance services to help participants to achieve their goals.

In the final part we have presented a lightweight infrastructure for agent coordination. This infrastructure is based on a P2P approach and then, inherits some of the advantages of these systems as a high degree of decentralization, self-organisation, low barrier to deployment (compared to client-server systems), organic growth, resilience to faults and attacks, and abundance and diversity of resources. We have exemplified this idea by presenting a P2P model for electronic institutions.

Acknowledgments
Work funded by projects EVE(TIN2009-14702-C02-01), AT (CSD2007-0022),

and the Generalitat of Catalunya grant 2009-SGR-1434. Marc Eseva enjoys a Ramon y Cajal contract from the Spanish Government.

References

1. N. R. Jennings, K. Sycara, M. Wooldridge, A roadmap of agent research and development, Autonomous Agents and Multi-agent Systems 1 (1998) 275–306.
2. Foundation for Intelligent Physical Agents, FIPA Abstract Architecture Specification, http://www.fipa.org/specs/fipa00001/ (2002).
3. J. F. Hübner, J. S. Sichman, O. Boissier, S-MOISE$^+$: A middleware for developing organised multi-agent systems, in: Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems, Vol. 3913 of LNCS, Springer, 2005, pp. 64–78.
4. M. Esteva, B. Rosell, J. A. Rodriguez-Aguilar, J. L. Arcos, Ameli: An agent-based middleware for electronic institutions, International Joint Conference on Autonomous Agents and Multiagent Systems 1 (2004) 236–243.
5. F. Bellifemine, A. Poggi, G. Rimassa, Developing multi-agent systems with JADE, Intelligent Agents VII Agent Theories Architectures and Languages LNCS 1986 (2001) 42–47.
6. A. Ricci, M. Viroli, A. Omicini, CArtAgO:A framework for prototyping artifact-based environments in MAS, in: Environments for Multi-Agent Systems III, Vol. 4389 of LNCS, 2006, pp. 67–86.
7. A. Omicini, A. Ricci, M. Viroli, Artifacts in the A&A meta-model for multi-agent systems, Autonomous Agents and Multi-Agent Systems 17 (3) (2008) 432–456.
8. R. Kitio, O. Boissier, J. F. Hübner, A. Ricci, Organisational artifacts and agents for open MAS organisations, in: Proceedings of COIN at AAMAS'08, Vol. 4870, 2008, pp. 171–186.
9. J. Vazquez-Salceda, W. W. Vasconcelos, J. Padget, F. Dignum, S. Clarke, M. Palau Roig, Alive: An agent-based framework for dynamic and robust service-oriented applications, in: Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010), 2010, pp. 1637–1638.
10. J. A. R.-A. Josep Ll. Arcos, B. Rosell, Engineering Autonomic Electronic Institutions, Engineering Environment-Mediated Multi-Agent Systems (2008) 76–87.
11. C. Zhang, S. Abdallah, V. Lesser, Integrating Organizational Control into Multi-Agent Learning, in: Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems, 2009, pp. 757–764.
12. J. Campos, M. Esteva, M. Lopez-Sanchez, J. Morales, M. Salamo, Organisational adaptation of multi-agent systems in a peer-to-peer scenario, Computing 91 (2011) 169–215.
13. A. Bogdanovych, Virtual institutions, Ph.D. thesis, University of Technology, Sydney, Australia (2007).
14. I. Seidel, Engineering 3d virtual world applications design, realization and evaluation of a 3d e-tourism environment, Ph.D. thesis, Technischen Universitat Wien Fakultat fur Informatik (2010).
15. T. Trescak, M. Esteva, I. Rodriguez, Vixee an innovative communication infrastructure for virtual institutions, in: Proceedings of The 10th International Conference on Autonomous Agents and Multiagent Systems, 2011, p. to appear.
16. J. Campos, M. Lopez-Sanchez, M. Esteva, Coordination support in mas, in: 8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 09), Budapest, Hungary, 2009, pp. 1301–1302.
17. R. Rodrigues, P. Druschel, Peer-to-peer systems, Communications of the ACM 53 (10) (2010) 72–82.