The Knowledge Engineering Review, Vol. 00:0, 1–24. © 2004, Cambridge University Press
DOI: 10.1017/S000000000000000 Printed in the United Kingdom

Adaptive Organizational Changes in Agent-Oriented Methodologies

Luca Cernuzzi[1] and Franco Zambonelli[2]

[1] *Departamento de Ingeniera Electrónica e Informtica Universidad Católica "Nuestra Señora de la Asunción" Campus Universitario, C.C. 1683, Asunción, Paraguay*
E-mail: lcernuzz@uca.edu.py
[2] *Dipartimento di Scienze e Metodi dell Ingegneria Università di Modena e Reggio Emilia Via Amendola 2, 42100 Reggio Emilia, Italia*
E-mail: franco.zambonelli@unimore.it

Abstract

We analyze the problem of modeling and developing multiagent systems from the organizational theory point of view. In particular, we focus on the critical issue of adapting multiagent systems organizations whenever changes in the structure of the multiagent system are required. We survey different relevant agent-oriented methodologies and discuss their suitability in dealing with adaptation in multiagent organizations. Finally, we present some critical considerations about the analyzed methodologies together with some open issues related to the process of modeling organizations for facilitating their adaptations.

1 Introduction

Different approaches for designing and building systems in complex and open environments have been proposed. Some of them focus on the methods, abstractions, and techniques normally used in traditional software engineering paradigms (i.e. component-based and functional-oriented) [Pressman, 2005; Sommerville, 2007]. More recent efforts try to take advantage from the *organizational theory*, and model/engineer complex software systems in term of components that participate in a computational organization with some specific role [Barber and Martin, 2001; Brooks and Durfee, 2003; Carley and Gasser, 1999; Horling et al., 2004; Matson and DeLoach, 2003; So and Durfee, 1998].

The organizational approach seems to be particularly interesting for *multiagent systems* (MAS), since these are a typically organized as a society of independent individuals, in need to interact with each other in order to reach a global or an individual goal. In fact, most approaches in the area of *agent-oriented software engineering* (AOSE) exploit organizational theory for the modeling and engineering of MASs. In MAS organizations, the types of interactions among component may vary a lot, and may change during time, depending on the goal of the system and the objective of each agent. Depending on the type of organization and on the perceived impact of the changes in the environment, adaptation is achieved by behavioral changes at agent level, modification of interaction agreements, or by the adoption of a new organizational structure. Therefore, some changes may affect the very structure of the MAS.

Whatever the software engineering paradigm adopted, one of the key problems for software engineers is dealing with changes and adaptations to the architecture of the software system. Software engineers are trying to anticipate the likely changes and adaptations which normally arise in almost all the software products after their deployments. However, these efforts can normally only tackle predictable changes that involve a reduced set of components (agents, in our case) but that do not influence the overall design of the system under construction. Thus, it is yet an open issue how to undertake continuous design changes/adaptations that may imply re-structuring the global organization of a system. The importance of the problem has been clearly recognized in previous literature. For example, in information systems studies [Bohem, 1981; Ghezzi et al., 1991], it was considered that the maintenance phase costs almost the 60% of the entire lifecycle of the system. Although there are no available specific studies of this kind for MASs, it is realistic to assume that MASs too will experience a similar trend, and possibly even more exacerbated as far as adaptive changes is concerned.

Unfortunately, although agents and MASs are often claimed as promising approaches for dealing with the dynamism of modern scenarios (i.e. with dynamic and open interactions and to interact in a dynamic environment), current AOSE discipline presents some limitations. A great deal of efforts in the AOSE area focuses on methodology definitions mainly trying to suggest a clean and disciplined approach to analyze, design and develop complex software systems based on the MAS paradigm, using specific methods and techniques [Ciancarini and Wooldridge, 2001; Zambonelli and Omicini, 2004]. However, AOSE methodologies typically promote the definition of static structure design for the overall organization of a MAS (i.e., for the roles to be played by the agents of a system and for the relations among these roles), and are not conceived to be ready for changes in the MAS organization after its deployment. In other words, an AOSE methodology should not only facilitate the development of a MAS, but should also facilitate *adaptive structural changes* in the overall organization of a MAS.

Being the organizational modeling a central point for the current AOSE methodologies, it is relevant to analyze how AOSE methodologies facilitate designers in coping with adaptive changes that may have a global impact on the overall design of a system (i.e., on the overall architecture/organization of MAS). Thus, the short term goal of this work is to highlight the advantages and limitations of AOSE methodologies, providing insight to overcome such limitations. At the same time, the long-term goal of this research is to reach a point where we will be able to develop and deploy MASs that autonomously self-adapt their behavior and re-structure their internal organization in response to contingencies.

Therefore, this study analyzes the problem of adaptive changes in MASs, paying special attention to the critical issues related to the choice of a general organizational structure, followed by a survey of some relevant AOSE methodologies to discuss their suitability in dealing with adaptation in MAS organizations. For this purpose, different aspects may have relevant impact. The application of principles like modularity and separation of concerns, the adopted process of the methodology, the explicit modeling of relevant abstractions for organizations (i.e. organizational structure and control regime), among other factors, may help designers to choose a different organization whenever circumstances claim for changes. Some critical considerations are presented about the analyzed methodologies. Moreover, we discuss some open issues regarding the processes of modeling organizations to facilitate their adaptations and notations to better capture all the relevant aspects of the organizational abstractions.

The paper is organized as follows. Section 2 focuses on the organizational issues including the organizational styles, the organization adaptation issues, the choice of specific organizational structure, and introduces a running example. Section 3 introduces the criteria upon which our critical survey is based, and analyses some relevant AOSE methodologies under the perspective of organizational changes. Section 4 concludes.

2 Organizational Abstractions and Adaptation

This section focuses on the main concepts of the organizational perspective and analyzes the key issues related to the organizational structures and the problems introduced by the need to adapt the structure of the MAS. Let us start with some relevant concepts.

Organizations can be generally defined as rationally structured instruments for the achievement of some stated goals [Selznick, 1948]. To such goal or some sub-goals it is necessary to identify the roles played within the organization. A role is the set of responsibilities and actions an actor assumes to fulfill some specific goals. Roles are always defined in relation to other roles. These role relationships form associations.

Organizational descriptions are abstractions at the level of role associations. The definition of roles and interdependencies among them is a first decomposition step to define the general *structure of the organization*. We use the term structure to identify the organization of the different components (mainly agents) of the system; that is, the groups of agents playing specific roles and the relationships among them.[1] The structure may be represented by means of a topology (a graph representation of the agents and their relationships). However, it is possible to organize actors (or agents) playing the defined roles in very different ways according to particular organizational styles [Ishida et al., 1992; Lesser, 1991; Carley and Gasser, 1999; Corkill and Lander, 1998; Kolp et al., 2005; Horling and Lesser, 2005], and depending on the circumstances. The organizational structure of MAS may impose constraints on the way the members of the organization interact with one another. Moreover, the structure can be explicitly designed or can be the result of emergent behavior. Usually, in the case of explicitly designed structures, there are goals for the society that must be reached through the interaction of the agents (desired behavior is external to the agents). On the other hand, in emergent MAS, the organizational global behavior, that is perceivable by an external observer, is the result of a bottom-up process based on interaction and local control decisions. In this case, the structure cannot be directly controlled by engineers or users.

In general, software systems are designed for a purpose [Horling and Lesser, 2005]. Thus a complete organizational description of a software system must include representation of purpose (global goal) and how this purpose is achieved through a goal-oriented control regime (norms and organizational rules) [Fox, 1981] for the whole system [Zambonelli et al., 2003]. Therefore, it is crucial to specify how goals are transmitted through the system, how the system changes due to changing requirements (in particular, changing goals) and environmental perturbations, and how it maintains its organizational integrity. Furthermore, in organizations it is necessary to take into account the interactions among different agents. Such interactions could represent the delegation of responsibility among roles and actors and the coordination/competition to achieve goals or sub-goals. Such interactions have to respect the rules governing the general behavior of the organization.

In summary, it is possible to state that an organization consists in two aspects: a structural aspect and an interactional aspect. The structural aspect, normally defined at design time, includes the partitioning of the whole system into groups of agents, the relationships among these groups and, for each group, the roles participating in with their relationships. The interactional aspect, normally specifying the runtime behavior, is related to the patterns of interactions and rules governing roles, agents, groups of agents and the overall organization; they are strongly related with the control regime. Therefore, modeling a MAS society amounts to consider the organizational structure of the systems (and probably each sub-organizational structure) as well as the control regime that governs its global behavior.

The main goal of this study is to analyze the critical issue of modeling the organizational structure and control regime of MAS societies and coping with changes of MAS organizations whenever circumstances claim for changes in the very MAS structure. In doing so, we are limiting this study to the problem of explicitly design the organizational structure without focusing on self-adaptive MASs. Nevertheless, we introduce considerations which can lead to more general insights also useful for self-adaptive systems. Therefore, the remaining of this section analyzes the kind of adaptations of the organizational structure that can emerge and the reasons for such

[1]The term *structure* may be easily related to *architecture* as usually used in the software engineering discipline (the architecture specifies in the global system the set of modules and their interrelationships). However, in agent-based computing the term architecture has sometimes been used to specify a specific type of agent (for example Believes-Desires-Intentions or BDI agents). Therefore, in order to avoid confusion, we prefer to use the term structure.

changes, and the problem of choosing the more adequate organizational structure for a MAS. In doing so, a running example supports the discussion on such issues.

2.1 Running Example: a Conference Management System

As a running example, let us consider an agent-based system for supporting the process of producing the technical program for an international conference [Cernuzzi and Zambonelli, 2005]. The process may be subdivided into three phases.

First, the submission phase: the program committee chair (PC-Chair) and the organizers prepare and distribute the call for papers. The authors submit their papers which are classified. A submission number is assigned to each paper and the authors are notified.

Second, the review phase: the PC-Chair distributes the papers among the PC-Members and collects their reviews. The PC-Chair decides on the acceptance/rejection of the papers, and then notifies the decision to the authors.

Third, the publishing phase: the PC-Chair prepares the conference program including all the accepted papers. The authors of the accepted papers have to produce a revised version of their papers and to send it to the publisher in charge of composing the proceedings.

Each phase involves different actors and may be supported by one MAS. In each MAS, software agents will be naturally associated to the actors involved in the process (authors, PC-Chair, PC-Members, reviewers) to support their work in the conference sub-organization. This may require agents to interact both directly with each other (according to patterns that will reflect the patterns of interactions in the real-world organizations), and indirectly (via exchanges of papers and review forms). Indeed, the types and roles of agents involved as well as the structure of the organizations and of inter-agent interactions naturally derive from the structure of the real-world organization. Nevertheless, for a conference, the overall structure of the real-world organization may dramatically vary from year to year for different reasons. Among them, the more relevant aspect is the number of submitted papers (which at some extent determines the size of the conference). Thus, the need of changing the structure of the management process may be forced by the need of keeping it manageable. This is particularly true for the reviewing phase, which involves a large number of actors, with different duties and variously interacting with each other.

Let us firstly assume that the conference organizers expect a limited number of submissions (*small size* conference), and then decide to organize the review process around a simple hierarchy in which the PC-Chairs distributes the papers among all the PC-Members and wait for the reviews.

Now let us assume that the number of submissions is higher than expected, reaching a *middle size* conference, which may be organized with a limited number of Co-Chairs (two or three instead of a single PC-Chair). At this point, Co-Chairs may decide to distribute among them the papers according to a peer-to-peer negotiation process before assigning each paper to PC-Members. The resulting structure may be considered a hybrid organization in which the upper level (corresponding to all the different Co-Chairs) adopts a peer-to-peer approach while the PC-Members are hierarchically subordinated to the Co-Chairs papers assignments.

Finally, let us assume that the number of submissions is much higher than expected (*large size* conference). At this point, the conference organizers may decide to adopt a different structure, i.e., a multilevel hierarchy, implying some change also in the underlying MAS supporting the process. In a multilevel hierarchy the PC-Chair will have to play a new previously unidentified role of ReviewPartitioner, to partition papers by areas and distribute each group to specifically appointed Vice-Chairs, each one in charge of handling papers in his/her area of competence.

Another interesting aspect in the example of Conference Management System, is that information about what the size of the conference will be (and thus about the most proper organizational structure to adopt) is generally available only a few days before the review process has to begin, i.e. when the papers are being submitted. This could force a dramatically fast re-structuring in the organization and, if the process is supported by a MAS, requires an extremely fast adaptation of the MAS structure. These problems, to different extents, are common in all those software systems devoted to support processes in an increasingly dynamic economy.

2.2 Adaptation

There exist different types of changeable systems which in general may be classified into two categories:

- *Adaptable System*: able to be modified by an external agent (with respect to a set of environmental states). In such systems, structural organization adaptation may be reached by adding, deleting or modifying their structural elements (e.g. roles, dependencies, norms, ontologies, communication primitives).
- *Adaptive System*: able to change itself by modifying its organizational design obtained by dynamic variation in emergent societies resulting from changes in the interaction between agents.

The changes in the organization are usually called *adaptive changes* (whether they occur within an adaptable or an adaptive system) and they may occur due to different reasons. Usually, environment changes and natural system evolution require adaptation of organizational structures. Moreover, MAS may also have to deal with changing goals. As a result, an adaptation of the organizational perspective in a MAS has to consider both the representation of the purpose and how this purpose is achieved through a control regime (norms and organizational rules).

Re-organization of organizations should therefore describe situations in which the operational behavior of the organization changes, due to changes in the social structure of the society – i.e. roles, relationships, norms or interactions patterns. In the example of the Conference Management System, such situation may arise for the unpredicted large number of submitted paper. Moreover, in this case we have to consider the problem of how can one MAS inspect and get feedbacks from the real-world organization to which it belongs in order to adapt accordingly.

In both cases, when the system needs to change its organization in order to meet radically changing goals or to adjust to changing environmental/resource constraints, this change can be achieved through the re-allocation of roles and associations.

Independently of the type of re-organization, since a MAS is subject to changes in its structure and control regime, it is necessary to find methods and techniques to model and manage the needed flexibility for such evolutionary systems. Several research efforts are being devoted to study and promote self-organization in complex software systems and, specifically, self-adaptive capabilities for MASs [Zambonelli and Omicini, 2004; Zambonelli et al., 2005]. These studies explore the possibility for complex MAS to either exploit adaptive self-organization phenomena or to promote self-inspect and self-reorganization in order to preserve specific functional and non-functional characteristics despite environmental contingencies. A number of algorithms and tools are becoming available, but the time for deployment of self-adaptive software systems and MAS has not yet come. In addition, it is worth outlining that, even if effective mechanisms of self-adaptation were available, the problem of having a MAS properly capture not only the internal needs of efficiency but also the external needs of the stakeholders remains open. How can one MAS inspect and get feedback from the real-world organization to which it belongs in order to adapt accordingly?

Moreover, adaptive systems concern emergent organization structures that cannot be predicted at design time and thus changes cannot be guided. In this case, the support of a methodology may be centered just on the documentation of the particular organization structures the MAS assumes in a specific moment. On the other hand, in adaptable MASs, the organization structure at a specific time is defined by the designer. This is a typical example of design decision that may be more naturally supported by almost any software engineering methodology, including the AOSE ones.

A possible way to anticipate such types of changes is to let the designers or agents to choose a specific organization structure among predefined catalogues of different organizations which may be adopted by the MAS. In this case, the creation of such catalog may be done using the abstraction, the models, and the notations proposed by an AOSE methodology. Nevertheless, it may be hard to really predict all possible situations.

Another possible way is to design self-adaptive MASs which are able to re-configure themselves according to the changes in the real-world organization structure. Also in this case, up to now it has been very hard to design systems whose behavior may evolve in a controlled and reliable way. Therefore, a reasonable compromise solution is to have a methodology that facilitates the design of a MAS with a different structure. Such solution is also to some extent a pre-request for future realizations of self-adaptive MASs.

2.3 Choosing the Organizational Structure

A relevant issue in the organizational perspective to deal specifically with adaptive changes is the choice of an appropriate *organizational structure* for a specific MAS. There are a number of (nearly orthogonal) forces that may drive the identification of an appropriate organizational structure. Such forces, considering the goals of the organization, include: the need to achieve organizational efficiency (or, equivalently, the need for the MAS to properly handle the computation and coordination complexity of a problem); the need to respect organizational rules (reflecting the control regime); and the need to minimize the distance from the real-world organization. All of these forces may contrast the desire to keep the design simple (for a more detailed discussion on the topic the reader should refer to [Zambonelli et al., 2003]). In the Conference Management System example the organizational rules may be quite stable (e.g. the number of reviews for each paper; the conflict of interest, etc.) while the organizational efficiency and the proximity to the real-world organization are strongly related to the number of submitted papers.

Accordingly, whenever the forces driving the design choice of the organization structure change (or whenever their relative weights change), the original design choice may become inadequate. Thus, some adaptation to the organizational structure may be required to preserve the functional and non-functional requirements compliance of the MAS.

The choice of the organizational structure issue introduces the question about who can do this. The appropriate structure can be decided by designers or specific agents with the responsibility to control the overall organization. In both cases, normally the structure is decided externally and imposed to the MAS. Meanwhile, in other systems that are able to change autonomously, the structure may be directly proposed by specific agents or may be selected from a catalogue of structures types pre-defined by designer or special agents. Hence, in both cases, the decision corresponds to agent/s of the MAS.

Another relevant issue is how to obtain the appropriate organizational structure. A possible way is by the continuous evolution of the structure self-controlled by agents and looking for a stable architecture. Likewise is the pre-designed evolution controlled by designers [Allen et al., 1997]. In those cases in which agents are in charge of making a decision, it is important that they have adequate skills to achieve this. In this sense, [Colman and Han, 2004] introduce the idea of the organizational management role, a specific role which is in charge of maintaining the representation or make a decision about restructuring the organization. In all the cases, an interesting issue is how AOSE methodologies support agent engineers in dealing with the adaptive changes, and the consequent problem of choosing an adequate organizational structure during the lifetime of the MAS. That is, it may be interesting to analyze how existing AOSE methodologies enable designers and developers to rapidly re-work the structure of a MAS in order for it to suit novel needs.

3 AOSE Methodologies and Adaptable MAS Organizations

This section focuses on how different existing AOSE methodologies deal with the need for modeling organizational structure in the case of adaptable MAS organizations.

The issue of continuous design change/adaptation organizations has been the subject of several studies. For example, an interesting work proposed by [Colman and Han, 2004] focuses on a role-oriented approach to describe the organizations and their adaptations. It is known that many of the actual AOSE methodologies recognize the concept of role as a first class abstraction for the design and modeling of MASs. However, the work of Colman and Han proposes an object-oriented role-based methodology while our main interest is devoted to MASs and their modeling.

In this area, other studies are specifically focused on the adaptation of MASs. One of the more interesting works is presented in [Dignum et al., 2004], which discusses re-organization aspects in different types of agent systems and concludes that a tradeoff between structure (predictability) and flexibility is needed to accommodate changes concerned with both adaptive and adaptable MASs. Other approaches concerns the agent generation at run-time in response to changes in requirements or in the environment, rather than the design of different organizational structures. In any case, non of the mentioned works focuses on the role of methodologies in the modeling of MAS under adaptive changes.

In summary, the specific role of the AOSE methodologies in the adaptation process has not yet received sufficient consideration in the specialized literature. Accordingly, with this regard we now analyze the key issues and survey how some representative AOSE methodologies tackles them.

3.1 A Design for Change Perspective to Support Changes in Organizational Structures

Agent engineers and developers need the support of AOSE methodologies that explicitly cover all the aspects related to the definition of an organizational structure and the associated control regime, and therefore facilitate possible re-organizations during the evolution of the system.

To promote adaptability in MASs, the AOSE methodology process needs to incorporate some interesting principles of software engineering, namely: *incrementality*, *modularity*, and *separation of concerns* (applied in this case to the organizational perspective). Such principles have been considered critical factors in other software engineering areas [Ghezzi et al., 1991; Rumbaugh et al., 1991; Booch, 1994; Meyer, 1997; Sommerville, 2007] and have to be applied both to the models and the process proposed by an AOSE methodology.

First of all, let us focus on the models. As we have seen in Section 2, any organization consists of structural and interactional aspects. Both aspects require to be explicitly specified during the design process. Therefore, in order to capture the structural aspect an AOSE methodology will have to consider models for partitioning the whole system into groups of agents, specifying the relationships among these groups and, for each group, the roles participating in with their relationships, and thus modeling the topology of the global MAS and of any sub-organization in the MAS. Moreover, for each role it is relevant to specify its goals and norms, which determine to a great extent the control structure.

On the other hand, to capture the interactional aspect an AOSE methodology will have to include models for the patterns of interactions and norms and rules governing roles, agents, groups of agents and the overall organization (that is, the control regime). The lack of an explicit model of the structure and of the control regime of organizations, makes harder for agent engineers to adopt pre-designed organizations structures and patterns available from a catalogue.

Complementary, relevant considerations have to be done on the processes proposed by the AOSE methodologies. Particularly, in the process of designing and developing a MAS, a methodology should clearly separate those aspects of the system that are intrinsic to the definition of the problem itself from those that derive from contingent choices based on the actual characteristics of the operational environment and/or the real-world organization. For example, in the Conference Management example, this means separating those functionalities and inter-dependencies intrinsic in a process of reviewing (e.g. functionalities of PC Chair and of reviewers, and protocols for sending back review forms) from those that instead derive from a specific contingent choice (e.g., separating the role of PC Member from that of the reviewers, and relying on paper bidding for assigning reviews). In that way, whenever unexpected changes occur, designers and developers are facilitated in identifying where to focus in order to restructure the MAS as needed without impacting on the whole system. This means that the analysis should not be affected by changes in the architectural/organization design.

Additionally, separation of concerns has to be applied in order to differentiate inter-agent organizational aspects from intra-agents ones. This separation may facilitate restructuring the MAS organization without having to re-think all of its roles and agents.

Finally, the process proposed by the adopted methodology may strongly influence the effectiveness of accommodating the needed adaptations. In effect, if a methodology prescribes too many steps to define an organizational structure or specify interactions patterns that influence the general behaviour of the MAS, this implies less flexibility for the adaptive changes.

Therefore, some critical factors for a AOSE methodology to promote adaptability in MAS development are:

- To prescribe the explicit modeling of the organizational structure (including sub-organizations) and control regime;
- To separate the functional requirement modeling from the organizational structure modeling;
- To differentiate intra-agents from the inter-agents organizational aspects;
- To adopt a flexible and incremental process;
- To exploit the modularity in order to facilitate re-use and to constrain the changes to the strictly necessary components;
- To promote a simple process (that does not prescribe too many steps and iterations to obtain the design models).

Taking into account such criteria, several AOSE methodologies proposed in the literature simply do not identify a clear separation between the intrinsic aspects of a MAS and the architectural aspects. For instance, methodologies such as Roadmap [Juan et al., 2002], Prometheus [Padgham and Winikoff, 2002], MaSE [DeLoach et al., 2001], AOR [Wagner, 2003], and DESIRE [Brazier et al., 2002], simply consider the organizational structure to stem in an implicit way from the identification of roles/agents and their interactions, without promoting any modularity and separation of concerns. Accordingly, they inherently have some problems in dealing with adaptations of MAS organizational structures.

Other proposed AOSE methodologies explicitly face the problem of structuring the organization of the MAS somewhat enforcing some degree of modularity and separation of concerns that make them more suitable for adaptable changes. In the following, we shall analyze some of the more recognized AOSE methodologies.

3.2 Gaia

The Gaia methodology [Zambonelli et al., 2003] explicitly focuses on using organizational abstractions to drive the analysis and design of multiagent systems. It models both the macro (social) aspect and the micro (agent internals) aspects of a MAS. Moreover, Gaia devotes a special effort to model the organizational structure and to specify the organizational rules that govern the global behavior of the agents in the organization avoiding conflict based on self-interest actions.

Gaia prescribes to clearly separate the analysis phase, where the basic characteristics of the system-to-be are captured and organized, from the architectural design phase, where all the results of the analysis are put at work to identify the most suitable organizational structure. The above clear separation, together with the specific structuring of the analysis phase and of its models, is an important factor to facilitate adaptive changes.

The result of the analysis phase in Gaia is very modular, clearly separating basic characteristics/functionalities of the systems, (i.e., the preliminary roles and interactions models) from characteristics of the operational environment (i.e., the environmental model) and from any additional constraints that the MAS will have to respect (i.e., the organizational rules). This implies that whenever contingencies calls for a re-thinking of some of the MAS specifications, the clear separation of concerns of the Gaia analysis models is likely to avoid global re-thinking of the whole analysis and, depending on the types of contingencies, promote a local tuning of a limited set of models. For instance, some functional change in how a sub-task is expected to be achieved will impact on the preliminary role model only; some changes in the global constraints that the MAS has to respect implies changes in the organizational rules only.

The prescriptions to explicitly model the organizational structure and to delay its identification to the architectural design phase are also of paramount importance. In fact, more than the outcome of the analysis, it is the choice of a specific organizational structure that is more likely to be affected by contingencies. Besides properly structuring the functional requirements of the analysis phase, the choice of a specific organizational structure has to take into account and is affected by a number of non-functional requirements and by various characteristics of the operational environment. Thus, whenever contingencies call for adaptive changes in the MAS, it is very likely that these contingencies will call for a new organizational structure, which in Gaia can be selected without globally affecting the design.

In fact, the analysis outcome of Gaia is a set of preliminary role and interaction models that exhibit no dependencies on a specific organizational structure. In the architectural design phase, after having chosen a specific organizational structure, the roles and interaction models can be finalized. Consequently, it is possible in the final roles and interaction models to clearly identify those which are intrinsic of the systems (i.e., those already identified from the analysis that may be considered as functional roles and interactions) from those that derive from the adoption of a specific organizational structure. Accordingly, whenever contingencies call for a new organizational structure, the designer is clearly facilitated in determining which parts of the system require some sort of re-design and which parts, instead, can be left unchanged. For example, if a designer is forced to switch from the small conference design to the large conference design, due to the modularity of Gaia models and the clear separation from analysis and architectural design phase, the designer can easily re-use all previously identified models of the analysis, re-applying them in the sub-hierarchies of Vice Chairs and PC Members, and introducing the new role of ReviewPartitioner to define the upper level of the hierarchy.

Thus, even if Gaia does not yet define any specific guideline for adaptive maintenance, its structuring of the development process somewhat facilitates adaptive changes, and also enables an effective re-use of previous experiences and models. In fact, an expert designer can easily apply known organizational structures possibly being supported by the availability of catalogues of organizational patterns in the context of a particular system, so as to more easily choose and specify a specific organizational structure for a MAS-to-be, and eventually easily re-shape the organizational structure of an existing system that requires some adaptation.

3.3 Tropos

The Tropos [Bresciani et al., 2004; Giorgini et al., 2005] framework aims at building agent-oriented software that operates within a dynamic environment. A curios thing is the Greek etymology of the word means easily changeable/adaptable. Tropos is mainly requirement-driven, adopting the$i*$ organizational modeling framework [Yu, 1995], and views the information systems as social structures, that is a collection of social actors, human or software, which act as agents, positions, or roles and have social dependencies among them.

Tropos supports four phases of software development mainly based on the actor, dependency, goal, and plan models. The early requirements analysis is concerned with the understanding of the problem through studying an organizational setting focusing on the intentions of stakeholders.

Late requirements analysis results in a requirements specification that describes all functional and non-functional requirements for the system to be. In the architectural design the system's global architecture, according to a specific architectural style, is defined in terms of subsystems, and interconnected through data and control flows that are modeled as dependencies. Finally, in the detailed design phase, additional details are introduced to define how the components present in the architectural model are going to fulfill its responsibilities according to design patterns.

All analysis of system requirements in Tropos is based on the goals that must be achieved by the system to be, and on the identification of the actors (whether humans or agents) that should be considered to achieved these goals. These requirements are modeled in the goal and actor diagrams. Once all goals are identified and assigned to specific actors, the design phase aims at producing the organizational structure of the systems (i.e., identifying relationships between agents), and at detailing the specific characteristics of the composing agents. To produce a software architecture, during architectural design Tropos concentrates on the key system actors, defined during late requirements analysis, and their responsibilities. These include the desired functionality of the system to be, as well as a number of quality requirements (Not Functional Requirements) related. Moreover, the analysis allows the evaluation of several alternative architectural styles, represented as operationalized softgoals, facilitating the designer in choosing a specific organizational structure.

An interesting aspect of Tropos is the definition of the organizational structure, which seems to be a key point for promoting adaptive organizational changes. Nevertheless, the topology (the organizational structure as we have referred in this paper) of the global organizational of the system to be is not explicitly modeled in Tropos. It is implicitly obtained by the dependences relations among goals of the different actors (including the system). Therefore, an explicit representation of the global structure, that is more likely to be affected by contingencies, may help agent engineers to accommodate adaptive changes.

On the other hand, some kind of changes in the control regime (i.e. those related to the interactions among agents) may be modeled by means of the actor and goal diagrams in term of changes in the dependencies.

The incremental and partially iterative process promoted by Tropos is somewhat interesting to facilitate changes at the design time and, at some extent, after implementation of the MAS. However, whenever contingencies call for a re-thinking of the global organization of the MAS, Tropos prescribes to return to actors and goal diagrams and possibly implies global re-thinking of the whole analysis effort which is based on these two main models. In effect, having the general goal of distributing the papers among reviewers in the case of a large size conference, the designers have to introduce the new role of ReviewPartitioner in the actor models and its associated objectives and plans in the goal model. This influence all the successive phases proposed in the Tropos process.

3.4 INGENIAS

INGENIAS [Gómez-Sánz and Pavón, 2003; Pavón et al., 2005] proposes a refinement approach that follows the Unified Software Development Process [Jacobson et al., 1999]. In INGENIAS, the general approach to specify MAS is to divide the problem in more concrete aspects that form different views of the system. Such meta-models allow the designers to incrementally define the architecture and the functionalities of the multiagent system, by focusing on different points of view during development. INGENIAS considers five different models (instances of meta-models that describe system views): Agent, Interaction, Tasks and Goals, Organization, and Environment.

During the analysis phase a meta-model acts as a guideline for the analyst, by indicating which entities have to be identified. In the design phase, the meta-model is refined, by identifying new components and relationships among them, in order to achieve the appropriate level of detail. The organizational viewpoint describes the framework where agents, resources, tasks, and goals coexist. It is defined by its structure, functionality, and social relationships. The organizational structure defines a decomposition of the multiagent system into groups and workflows. Such general structures are enriched by the workflows, related to the way to cover the functionalities, and the social relationships which can be established at different levelsbetween organizations, groups, agents, or roles. Social rules state restrictions on the interactions between entities in the organization.

Complementary, the interaction model describes how interaction among agents takes place with special emphasis on the protocol that follows the interaction. Such relationships clarify organization behavior and specify which constraints exists in the interaction among agents. Nevertheless, it is not clear if it is possible to specify rules which prevent malicious behaviors. Therefore, we consider that the modeling of the organizational structure is adequately covered and that the social relationships of dependency complemented with the protocols specification cover the main aspects of modeling the control regime but the general organizational rules.

The incremental approach adopted by INGENIAS makes it somewhat suitable for accommodate possible changes arising during the design and the development process. Whenever contingencies call for a re-thinking of some of the MAS specifications, the separation of the modeling activities in different views may facilitate to avoid global re-thinking of the whole solution. Specifically, possible adaptive changes are more likely to affect the choice of a specific organizational structure which modeling is strongly related with the organization and interaction models. The organizational view describes how the system components are grouped together, which are the interaction among the agents and considers the satisfaction of a set of goals. In the analysis-inception phase, organization models are produced to sketch how the MAS looks like (the MAS architecture) with an organizational model. This result is refined in the analysis-elaboration phase to identify common goals of the agents and relevant tasks to be performed by each agent. In the design-elaboration phase designers add workflows among the different agents to improve the organization model. Finally, in the design-construction phase, social relationships of dependency (in form of subordination and client-server relationships) are defined. Considering the scenarios of the running example, to deal with the change in the structure of the organization, designers have to re-do a lot of workflows in order to obtain the new models. In effect, it implies to pass again from the analysis-inception (with the sketch of the new MAS architecture) up to the design-elaboration (with the definition of dependencies) phases to obtain a modified organizational model. Moreover, this approach would be limited by the inherent dependencies among different views.

As we have seen, the process of designing and developing multiagent systems with INGENIAS is well structured, incremental, and considers complementary perspectives in the high number of workflows it proposes. At the same time, the process is quite complex due to the many activities the designers have to do in order to produce the different models before the developing stage. Hence, for the case of adaptive changes claimed by different circumstances, the process proposed by INGENIAS tends to reduce the facility to easy accommodate such changes.

3.5 PASSI

The PASSI methodology [Cossentino and Sabatucci, 2004] is based on several years of experiences in modeling and developing robot systems and with the aim of designing MAS in an iterative-incremental approach.

PASSI is constituted of five main models (System Requirements, Agent Society, Agent Implementation, Code, and Deployment) and facilitates a clear separation in perspectives of the design process of MAS. The five perspectives considered are:

- The Architectural perspective: abstraction of the whole system as a set of functionalities and their logical implementation;
- The Social perspective: to capture the society of agents that interact to reach the goals of the system;

- The Knowledge perspective: focusing on each single agent, its functional and behavior details;
- The Resource perspective: oriented towards the reuse of patterns of agents and their tasks;
- The Computer perspective: considering the spreading of the files that constitute the agent based system and the influence of the available hardware platform.

Clearly, the two main perspectives related with the adaptable organizational issues above mentioned are the Social and the Architectural ones.

The Architectural perspective covers the system requirements and the agent implementation models. However, the first one only focuses on the functional description of the system, starting from the traditional use-case modeling with scenarios specifications, and then identifying the agents and specifying their roles and tasks, without explicitly considering the goals of the MAS. The Agent Structure Definition phase captures the general relationships among agents of the system by means of a class diagram. To some extent, such diagram is related with the organizational structure of the MAS but it does not cover adequately relevant aspects as the topology. Consequently, it may be not so easy to accommodate possible adaptive changes in the organizational structure during the evolution of the MAS.

In the Social perspective, some complementary aspects of the relationships and dependencies among agents (taking into account their roles) are captured in the Role Description phase. The authors state that in the same phase the social rules and the behavioral laws of the society of agents may be expressed; however in the bibliography we have not found available examples. Moreover, the interaction among agents is covered in the Protocol Description phase and the communications aspects are partially captured in the Communication Ontology Description diagram. Also, relevant aspects of each single agent are captured by means of the Agent Behavior Description phase. However, in your best known, no stress is devoted to the global organizational structure nor to the general rules governing the overall system.

Considering the PASSI process, incrementality towards different iterations can in general help in accommodating changes during the design time. However, if adaptations arise after the deployment of the MAS, the great number of phases in a cycle may require a long time before accommodating the changes in a new prototype code. As an example, for distributing the papers among reviewers in the case of a large size conference, the designers have to introduce the new role of ReviewPartitioner. This change does not affect the Agent Identification model (it holds the agent PC-Chair with different instances for the General Chair and the Vice-Chairs) , but agent designers have to define a new role in the Role Identification step which is part of the System Requirements Model and the Agent Society Model. Moreover, the Role Description, the Tasks Specification, the Agent Structural Definition, the Agent Behavior Description (including the multiagent description and the single-agent description diagrams) phases need to be modified according.

Therefore, we consider some limitations for adaptive changes. The first one is that in PASSI not all the aspects of organizational structure are explicitly modeled (or at least the lack of enough information on the specification of the control regime). And the second one would be the vast number of steps and models that need to be modified to accommodate changes to the global MAS organization. Nevertheless, an Agile version of PASSI has been proposed [Chella et al., 2004] which preserves the iterative and incremental nature but by following the ideas of agile processes leads to a quicker process more oriented to code delivery than to documentation production.

3.6 OMNI

OMNI (Organizational Model for Normative Institutions) [Dignum et al., 2005] is a framework for modeling different types of MAS ranging from closed to open flexible systems, especially emphasizing the organizational and normative perspective. This framework is composed of three dimensions, namely, Normative, Organizational and Ontological. OMNI considers three different levels of abstraction: abstract (similar to the first step of the requirement analysis), concrete (analysis and design process) and implementation (in a given multiagent architecture). For our purpose we focus the discussion on the concrete level which is strongly related to the analysis and design of MAS.

The concrete level of the Organizational dimension specifies an Organizational Model by means of the social structure and the interaction structure. The social structure consists of roles and groups (set of roles) and is represented in terms of a role hierarchy with the specification of role dependencies (bidding, request and delegation). These characteristics may help to better capture the structure of the MAS. The interaction structure specifies the transition relationship between a set of scenes. Each scene describes the roles involved, the interaction pattern, the desired results, and the norms regulating the interaction.

These structures need to be connected with the role norms, scene norms and transition norms, defined in the normative dimension, and with the ontologies and communication languages defined in the Ontological dimension.

In the Normative dimension abstract norms are iteratively refined into more concrete norms, which are then translated into the rules, violations and sanctions that implement them. Such dimension covers the control regime specification. Finally, in the Ontological dimension OMNI describes both the content (domain ontologies) and the communication acts that define the interactions languages used in the organizational model.

The structure of the organization is explicitly modeled in the Organizational Model according to predefined types of society (market, hierarchy or network) and the identification of the basic roles and of the interaction structure. Being OMNI proposed for open societies of agents, the structure model is partially reduced to the predefined types of society. Complementary, one of the strong points of OMNI is the normative dimension which covers the norms and rules for both agents and the organization and, in addition, the violations cases and corresponding sanctions for such norms. Consequently, in this dimension it is easy to accommodate adaptive changes.

On the other hand, the process implicitly adopted by OMNI applies the modularity principle and makes a clear separation of the organizational structure from the control regime (normative dimension) perspectives. Moreover, it allows the designer to model them in an incremental way during the concrete level of abstraction. Considering the running example, to accommodate the structural changes for the case of switching form a small to a large size conference the designers have to introduce the new role of ReviewPartitioner and some interactions. Both are considered in different steps of the Organizational Model and imply changes into the Role Hierarchy graph, and the Role Dependencies graph. Nevertheless, OMNI does not prescribe separation between the functionalities and the organizational structural design which is obtained by associating objectives and roles and decomposing them into sub-roles (both are jointly carried on in the concrete level) and between inter-agent from intra-agent organizational aspects. Thus, when adaptive changes call for modification in the organizational structure, designers have to devote a consistent effort to adapt the overall organizational dimension at the concrete level, and particularly the role hierarchy as well as the role dependencies models.

3.7 Discussion and Open Issues

Current methodologies for MASs design focus on the analysis of the initial requirements and their consequences for MAS design, but do not often provide explicit guidelines for reorganization and adaptation of the organizational structure later on. As we have seen, to facilitate the accommodation of adaptive changes in the MAS organizations different aspects have to be considered.

Two directly related abstractions that are necessary in modeling the organizational perspective are the organizational structure and the control regime [Fox, 1981] that govern the general behavior of the MAS. As discussed in the previous sub-sections, whenever methodologies do not prescribe to explicitly model organizational structures and control regime, it can be more difficult to accommodate their probable changes. All the surveyed methodologies cover those aspects to different extents. More specifically, they cover the organization structure specification but PASSI and Tropos would need to explicitly model the topology to better help agent engineers to accommodate adaptive changes. In the control regime specification Gaia and INGENIAS offer the most interesting approaches, being very promising the PASSI one (but agent researchers and engineers need more clear documentation and examples).

Furthermore, as stated in Section 3.1, the application of some interesting principles strongly adopted in traditional software engineering may facilitate the building of adaptable and adaptive MAS. Among them, design-for-change, incrementality, modularity, and separation of concerns seems to be very relevant. Each methodology analyzed so far covers in different way such aspects. Gaia prescribes a good separation of concerns, specially focusing on a separation between the general functional and not functional requirements and the organizational ones. However, its incremental process is based on a waterfall like model and consequently is not iterative. Tropos promotes an incremental and partially iterative process that is somewhat interesting to facilitate adaptive changes of the MAS. Still, the separation of concerns for the organizational perspective is not so clear and possibly implies global re-thinking of the whole analysis to accommodate adaptive changes. INGENIAS adopts an incremental approach and the separation of the modeling activities into different views which may facilitate to reorganize adaptable MAS. Nevertheless, the process proposed by INGENIAS implies to pass again from the analysis-inception up to the design-elaboration phases to obtain a modified organization model. PASSI is intrinsically incremental-iterative driven but prescribes a great number of phases in the process cycle which implies a great effort to accommodate changes. Moreover, PASSI facilitates a clear separation in perspectives of the design process of MAS but we consider a limitation for adaptive changes that not all the aspects of organizational structure are explicitly modeled.

Others general considerations can be made on the design and construction process for adaptable MAS. A first consideration is that most of the methodologies are concerned with the analysis and design processes only [Cernuzzi et al., 2005]; few are trying to cover the development and deployment of the system; fewer yet are concerned with the maintenance stage of the system. Thus, even when a methodology is more suitable for a design-for-change perspective, a specific attention to the maintenance process and the definition of proper guidelines for change and adaptation are lacking, which is a great limitation for modern methodologies. It is also worth outlining that the dynamism of modern scenarios and the need for nearly continuous adaptive changes makes the traditional waterfall software process model, upon which most methodologies explicitly or implicitly rely, very unsuitable [Cernuzzi et al., 2005]. Evolutionary process models and, more specifically, agile extreme process models may better facilitate engineers in the design and building of adaptable MAS. However, current agile and extreme software process models focus on small- to medium-size projects, and they are not yet ready to tackle the complexity of developing large-scale adaptable and adaptive MAS.

Despite such considerations, different open issues may be envisioned for methodologies to better capture the organization perspective in MAS design. One of them is related to the notations adopted for modeling the organizational structures. In our opinion, this is an open issue for all the methodologies. The challenge is the need to combine the advantage of organizational theory notations with the representation of MAS abstractions. In this sense, an interesting suggestions may be offered by the Agent Modeling Language AML approach [Cervenka et al., 2006] which devotes special attention to the social/organizational aspects introducing different diagrams to capture the social structure, the social behavior and the social attitudes. One of its main contributions is its powerful notation being specified as a conservative extension of UML 2.0. Still, AML is a modeling language more than a complete methodology that, perhaps, may be integrated in a more organic and structured approach as offered by some existing or new agent oriented methodology.

Another interesting open issue is related to the different organizational styles (i.e., the way to describe a type of organization by using specific organizational abstractions and notations). In effect, we know the structure may influence the resource access and allocation, the coordination pattern, the authority relationships, the data flow among agents as well as other system characteristics. However, more efforts are necessary to better understand the influence of the organizational styles on the MAS organizational structure (and the adaptability for eventual changes), and on the organizational patterns.

4 Conclusions

In this paper, we analyzed the critical issue of dealing with changes that may require re-structuring the global organization of a MAS. More specifically, we have considered the role of AOSE methodologies in supporting the modeling of the adaptive changes to the organizational structure. Therefore, we presented a discussion about how different relevant AOSE methodologies (namely, Tropos, INGENIAS, PASSI, OMNI and Gaia) can to some extent facilitate engineers in tackling the likely changes that will appear in the organization of a MAS after its deployment.

As we have shown, different factors may facilitate the accommodation of adaptations in MAS organizations whenever circumstances call for changes in the very MAS structure. First of all, it is extremely important for AOSE methodologies to explicitly model the organizational structure and control regime. Moreover, the application of principles like design-for-change, incrementality, modularity, and separation of concerns may facilitate the building of adaptable and adaptive MAS. Finally, the process proposed by a methodology may strongly influence the level of flexibility to ease reach adaptation in MAS organizations. In this sense, all the existing AOSE methodologies may take advantage from the discussion of this study to adapt their models and processes to better deal with adaptive MASs.

Our current research work is focused on proposing more specific guidelines and conceptual tools to support MAS engineers to deal with adaptations of MASs. An additional issue that we consider very important relates to adaptation at the implementation level, i.e., how does changes in the design reflect in the implementation, and what different problems may arise at this level that we have still not identified? The final long-term goal of these studies is to eventually reach a point in which we will be able to develop and deploy MASs that autonomously self-adapt their behaviour and re-structure their internal organization in response to contingencies.

References

Allen, R., Douence, R. & Garlan, D. 1997 *Specifying Dynamism in Software Architectures.* Proceedings of ACM Workshop on the Foundations of Component-Based Systems. ACM Press, New York (NY).

Barber, K. S. & Martin, C. E. 2001 *Dynamic Adaptive Autonomy in Multiagent Systems: Representation and Justification.* Journal of Pattern Recognition and Artificial Intelligence, 15(3), 405-433.

Boehm, B. 1981 *Software Engineering Economics.* Prentice-Hall, Englewood Cliffs (NJ).

Booch, G. 1984 *Object Oriented Analysis and Design with Applications.* The Benjamin-Cummings Publishing Company, New York (NY).

Brazier, F., Jonker, C. & Treur, J. 2002 *Principles of Component-Based Design of Intelligent Agents.* Data and Knowledge Engineering, 41(2), 1-28.

Bresciani, P., Giorgini, P., Giunchiglia, F., Mylopoulos, J. & Perini, A. 2004 *TROPOS: An Agent-Oriented Software Development Methodology.* Journal of Autonomous Agents and Multi-Agent Systems, 8(3), 203236.

Brooks, C. & Durfee, E. 2003 *Congregation Formation in Multiagent Systems.* Journal of Autonomous Agents and Multiagent Systems, 7(1-2), 145170.

Carley, K. M. & Gasser, L. 1999 *Computational Organization Theory.* In: Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, The MIT Press, Cambridge (MS), 299330.

Cernuzzi, L., Cossentino, M. & Zambonelli, F. 2005 *Process Models for Agent-based Development.* Journal of Engineering Applications of Artificial Intelligence, 18(2), 205222.

Cernuzzi,L. & Zambonelli, F. 2006 *Dealing with Adaptive Multi-Agent Organizations in the Gaia Methodology.* Proceedings of the 6th International Workshop on Agent-Oriented Software Engineering, LNCS Vol. 3950, Springer Verlag, Berlin (D), 109-123.

Cervenka, R., Trencansky, I. & Calisti, M. 2006 *Modeling Social Aspects of Multi-Agent Systems: the AML Approach.* Proceedings of the 6th International Workshop on Agent-Oriented Software Engineering, LNCS Vol. 3950, Springer Verlag, Berlin (D).

Chella, A., Cossentino, M., Sabatucci, L. & Seidita, V. 2004 *From PASSI to Agile PASSI: Tailoring a Design Process to Meet New Needs.* IEEE/WIC/ACM International Joint Conference on Intelligent Agent Technology (IAT-04), Beijing (China).

Ciancarini, P. & Wooldridge, M. J. 2001 *Agent-Oriented Software Engineering.* In: Agent-Oriented Software Engineering, LNAI Vol. 1957. Springer-Verlag, Berlin (D), 1-24.

Colman, A. & Han, J. 2004 *Organizational Abstractions for Adaptive Systems.* Technical Report No: SUTIT-TR2004.03/SUT.CeCSES-TR003, School of Information Technology, Swinburne University of Technology.

Colman, A. & Han, J. 2005 *Coordination Systems in Role-Based Adaptive Software.* International Conference on Coordination Models and Languages, LNCS Vol. 3454, Springer Verlag, Berlin (D), 63-78.

Corkill, D. D. & Lander, S. E. 1998 *Diversity in Agent Organizations.* Object Magazine, 8(4), 41-47.

Cossentino, M. & Sabatucci, L. 2004 *System Implementation.* In: Agent-Based Manufacturing and Control Systems: New Agile Manufacturing Solutions for Achieving Peak Performance, CRC Press.

DeLoach, S., Wood, M. & Sparkman, C. 2001 *Multiagent Systems Engineering.* International Journal of Software Engineering and Knowledge Engineering, 11(3), 231-258.

Dignum, V., Sonenberg, L. & Dignum, F. 2004 *Dynamic Reorganization of Agent Societies.* Proceedings of the Workshop on Coordination in Emergent Agent Societies CEAS at ECAI, Valencia (S).

Dignum, V., Vázquez-Salceda, J. & Dignum, F. 2005 *OMNI: Introducing Social Structure, Norms and Ontologies into Agent Organizations.* Programming Multi-Agent Systems: Second International Workshop, LNAI Vol. 3346, Springer Verlag, Berlin (D), 181-198.

Fox, M. S. 1981 *An Organizational View of Distributed Systems.* IEEE Transactions on Systems, Man, Cybernetics, 11(1), 70-80

Ghezzi, C., Jazayeri, M. & Mandrioli, D. 1991 *Fundamentals of Software Engineering.* Prentice Hall International, Upper Saddle River (NJ).

Giorgini, P., Kolp, M., Mylopoulos, J. & Castro, J. 2005 *Tropos: A Requirements-Driven Methodology for Agent-Oriented Software.* In: Agent-Oriented Methodologies, Idea Group, Hershey (PA), 20-45.

Gómez-Sanz, J. & Pavón, J. 2003 *Agent Oriented Software Engineering with INGENIAS.* Proceedings of the 3rd Central and Eastern Europe Conference on Multiagent Systems, Springer Verlag, LNCS Vol. 2691, Springer Verlag, Berlin (D), 394-403.

Horling, B. & Lesser, V. 2005 *A Survey of Multi-Agent Organizational Paradigms.* The Knowledge Engineering Review, 19(4), 281-316.

Horling, B., Mailler, R. & Lesser, V. 2004 *A Case Study of Organizational Effects in a Distributed Sensor Network.* Proceedings of the AAAI-04 Workshop on Agent Organizations: Theory and Practice, San Jose (CA).

Ishida, T., Gasser, L., & Yokoo, M. 1992 *Organization Self-design of Distributed Production Systems.* IEEE Transactions on Knowledge and Data Engineering, 4(2), 123-134.

Juan, T., Pearce, A. & Sterling, L. 2002 *ROADMAP: Extending the Gaia Methodology for Complex Open Systems.* Proceedings of the First International Conference on Autonomous Agents and Multi-Agent Systems, Bologna (I), 3-10.

Kolp, M., Giorgini, P. & Mylopoulos, J. 2006 *Multi-Agent Architectures as Organizational Structures.* Journal of Autonomous Agents and Multi-Agent Systems 13(1), 3-25.

Lee, J. S. & Bae, D. H. 2002 *An Enhanced Role Model for Alleviating the Role-binding Anomaly.* Software: Practice and Experience, 32(14), 1317-1344.

Lesser, V. 1991 *A Retrospective View of FA/C Distributed Problem Solving.* IEEE Transactions on Systems, Man, and Cybernetics, 21(6),1347-1363.

Lesser, V. 1998 *Reflections on the Nature of Multi-Agent Coordination and Its Implications for an Agent Architecture.* Journal of Autonomous Agents and Multi-Agent Systems 1(1), 89-111.

Matson, E. & DeLoach, S. 2003 *Using Dynamic Capability Evaluation to Organize a Team of Cooperative Autonomous Robots.* Proceedings of the International Conference on Artificial Intelligence (IC-AI03).

Meyer, B. 1997 *Object-Oriented Software Construction (2nd edition)*, Prentice-Hall, Englewood Cliffs (NJ).

Padgham, L. & Winikoff, M. 2002 *Prometheus: A Methodology for Developing Intelligent Agents.* Proceedings of the First International Conference on Autonomous Agents and Multi-Agent Systems, Bologna (Italy), 135146.

Pavón, J., Gómez-Sanz, J. & Fuentes, R. 2005 *The INGENIAS Methodology and Tools.* In: Agent-Oriented Methodologies, Idea Group, Hershey (PA), 236-276.

Pressman, R. 2005 *Software Engineering. A Practitioners Approach (6th Edition).* McGraw-Hill, New York (NY).

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. & Lorensen, V. 1991 *Object-Oriented Modeling and Design.* Prentice-Hall, Englewood Cliffs (NJ).

Selznick, P. 1948 *Foundations on the Theory of Organizations.* American Sociological Review, 13(1), 25-35.

So, Y. & Durfee, E. 1998 *Designing Organizations for Computational Agents.* In: Simulating Organizations, The MIT Press, Cambridge (MS), 47-64.

Sommerville, I. 2007 *Software Engineering (8th Edition).* Pearson Education Limited, Harlow (UK).

Wagner, G. 2003 *The Agent-Object-Relationship Metamodel: Towards a Unified View of State and Behavior.* Information Systems, 28(5), 475-504.

Wooldridge, M. J., Jennings, N. R. & Kinny, D. 2000 *The Gaia Methodology for Agent-Oriented Analysis and Design.* Journal of Autonomous Agents and Multi Agent Systems, 3(3), 285-312.

Yu, E. 1995 *Modelling Strategic Relationships for Process Reengineering.* Ph.D. Thesis, Department of Computer Science, University of Toronto (CA).

Zambonelli, F., Wooldridge, M. J. & Jennings, N. R. 2003 *Developing Multiagent Systems: The Gaia Methodology.* ACM Transaction on Software Engineering and Methodology 12(3), 417-470.

Zambonelli, F. & Omicini, A. 2004 *Challenges and Research Directions in Agent-Oriented Software Engineering.* Journal of Autonomous Agents and Multiagent Systems, 9(3), 253-283.

Zambonelli, F., Gleizes, M. P., Mamei, M. & Tolksdorf, R. 2005 *Spray Computers: Explorations in Self-organization.* Journal of Pervasive and Mobile Computing, 1(1), 1-20.