Using the $\mathcal{M}$OISE$^+$ Model for a Cooperative Framework of MAS Reorganization*

Jomi Fred Hübner[1] **, Olivier Boissier[2], and Jaime Simão Sichman[1] ***

[1] LTI / EP / USP
Av. Prof. Luciano Gualberto, 158, trav. 3
05508-900 São Paulo, SP
{jomi.hubner,jaime.sichman}@poli.usp.br
[2] SMA / SIMMO / ENSM.SE
158 Cours Fauriel
42023 Saint-Etienne Cedex, France
Olivier.Boissier@emse.fr

Abstract. The work reported here proposes a characterisation of the reorganization problem for a MAS in cases where the agents themselves try to maintain the organization well adapted to both environmental changes and their purposes. The main problems of the reorganization process are thus distributed in four phases: monitoring (when to reorganize), design (ways of building a new organization), selection (how to choose an organization), and implementation (how to change the current running organization). We show how we could use $\mathcal{M}$OISE$^+$ to specify this process and thus turn to an organization centered point of view of the reorganization in MAS.

1 Introduction

In an organizational centered (OC) point of view [9], the organization of a Multi-Agent System (MAS) can be seen as a set of constraints that a group of agents adopts in order to easily achieve their social purposes. The Fig. 1 briefly shows how an organization could explain or constrain the agents' behavior in case we consider an organization as having both *structural* and *functional* dimensions. In this figure, it is supposed that a MAS has the purpose of maintaining its behavior in the set P, where P represents all behaviors which draw the MAS's social purposes. In the same figure, the set E represents all possible behaviors in the current environment. The MAS's organizational structure is formed, for example, by roles, groups, and links that constrain the agents' behavior to those inside the set S, i.e., the set of possible behaviors ($E \cap S$) becomes closer to P. It is a matter of the agents, and not of the organization, to conduct their behaviors from a point in $((E \cap S) - P)$ to a point in P. In order to help the agents in this task, the functional dimension contains a set of global plans F that has been proved efficient ways of turning the P behaviors active.

* This work was partially developed during the first author stay at ENSM, France.

** Supported by FURB, Brazil; and CNPq, Brazil, grant 200695/01-0.

*** Partially supported by CNPq, Brazil, grant 301041/95-4; and by CNPq/NSF PROTEM-CC MAPPEL project, grant 680033/99-8.

Being well organized is a valuable property of a MAS, since it helps the system to assure its efficacy and efficiency [3]. Our general view of the organization for a MAS, depicted in the Fig. 1, allows us to state a minimal condition for a MAS to be well organized: $E \cap S \cap F \cap P \neq \emptyset$, i.e., the behaviors which lead to the social purpose achievement are allowed by the organization. However it is almost impossible (indeed undesirable) to specify an organization where the allowed agents' behaviors fit exactly the set P, since this set also depends on the environment, i.e., the behaviors to achieve the social purpose are conditioned by the environment. Different environments require different sets of P behaviors. Moreover, if the sets S and F are too small, the MAS will have adaptation problems when environmental changes happen due to the extinction of the agents autonomy by the organization. On the other side, if S and F are too big, the organization will not be effective since the agent's behaviors are not sufficiently constrained.

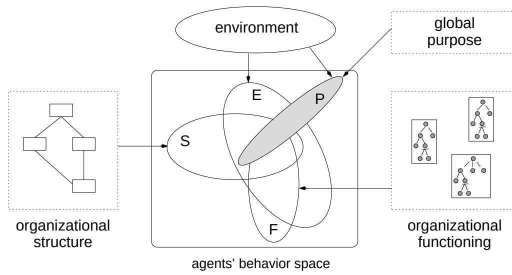


Fig. 1. The organization effects on a MAS

Identifying a good size for the set of organizational allowed behaviors is indeed another way of conceiving one important MAS problem which is how to conciliate collective constraints with the agent autonomy. Normally MAS methodologies are concerned with this problem ([17], to cite some). However, even if the MAS has a good organization, dynamic changes either in the environment or global purposes may cause the looseness of this property. Moreover, if we consider the organization unchangeable, the agents which have several experience and information about the organization, can not contribute to its adaptation. They loose the autonomy regarding its organization, i.e., regarding the set of constraints imposed over them. In other words, this problem could be expressed as how to conciliate an agent centered point of view (AC) point of view with an OC point of view. This situation brings the *reorganization* problem up, i.e., how the agents themselves might change the current organization [12].

If we assume that (i) there is no better organization for a context [2] and (ii) different organizations will give different performances for a system [3], a MAS needs to be capable of reorganizing itself in order to well suit in its environment and to efficiently achieve its goals. Both the importance that this adaptation feature has for a MAS and the need to understand how this process occurs have motivated the work reported here. Our objective is therefore to propose a general process for the reorganization (Sec. 3) and its specification (Sec. 4) based on the $\mathcal{M}$OISE$^+$ model (Sec. 2). We will thus show how the reorganization itself could be expressed and controled in an OC point of view.

2 The $\mathcal{M}$OISE$^+$ organizational model

The $\mathcal{M}$OISE$^+$ (Model of Organization for multI-agent SystEms) follows the general view of the organization depicted in the Fig. 1 and therefore considers the organizational structure and functioning. However, this model adds a deontic relation among these first two dimensions to better explain how a MAS's organization collaborates for the social purpose. These three dimensions form the Organizational Specification (OS). When a set of agents adopts an OS they form an Organizational Entity (OE) and, once created, its history starts and runs by events like other agents entrance, agents leaving the OE, group creation, role adoption, mission commitment, etc [6].

The $\mathcal{M}$OISE$^+$ structure, called Structural Specification (SS), is built in three levels: (i) the behaviors that an agent is responsible for when it adopts a role (*individual* level), (ii) the interconnections between roles (*social* level), and (iii) the aggregation of roles in large structures (*collective* level). The $\mathcal{M}$OISE$^+$'s SS allows us to ascribe the well formed attribute to a group in case the roles of one agent are compatible, the minimum and maximum number of role players are satisfied by the group, etc.

The Functional Specification (FS) describes how a MAS usually achieves its *collective* goals [1], i.e., how these goals are decomposed into plans and distributed to the agents by missions, on which preferences may be set. Such decompositions are called Social Scheme (SCH) which may be set either by the MAS designer who specifies its expertise in a SCH form or by the agents that store their past (best) solutions (as an enterprise does through its "procedures manual").

The organization's structure and functioning are linked by the Deontic Specification (DS) which states the roles' permissions and obligations for missions. This linkage allows the MAS to change the SS without changing the FS, and vice versa, the system only needs to adjust its DS relation.

The main property of the $\mathcal{M}$OISE$^+$ is to be an OC model [9] where the first two dimensions can be specified almost *independently* of each other and after properly linked by the deontic dimension. Despite some similarities among the $\mathcal{M}$OISE$^+$ concepts and the object oriented area, there is not a "new Role(x)" command to create an agent for a role. In our point of view, the agents of a MAS are autonomous and decide to "follow" the rules stated by the OS. They are not created by/from the organization specification, instead they just accept to belong to groups playing roles.

3 A general view of the reorganizational process

In the OC point of view, the organization is usually built upon two core concepts: (i) the description of the organization (types of roles, groups, links, global plans, etc. — OS in the $\mathcal{M}$OISE$^+$) and (ii) the current state of one instance of this organization type formed by agents with a common social purpose (OE in the $\mathcal{M}$OISE$^+$). This point of view allows one to define reorganization as a process which changes the current state of one of these two dimensions into a new one. Notice that there is a wide spectrum of change types. It can be, for instance, the adoption of a role by an agent (which changes only the OE) or a change in some group's set of roles (a change in the OS). We consider this spectrum as divided in two levels of organizational changes:

OE level : comprises changes in the roles/missions the agents are playing, the creation of groups, etc. (e.g. [4,7,13]); and

OS level : comprises changes in the organizational specification. In this level, the $\mathcal{M}$OISE$^+$ let us to consider three sub-levels of reorganization:

– SS changes in the groups, roles, and links,
– FS changes in the SCHs, missions, and preferences, and
– DS changes in the obligation and permission anchors of missions to roles.

For example, in [14] the agents can choose among three organization types: market, hierarchy, or community (SS level). In [5] the OS is described by TÆMS, thus the agents may change the task descriptions (FS level). In [12] the OS is considered as a task decomposition tree, reorganization is therefore to change the branching factor (FS level). In the [13], the OS represents a soccer team formations with roles representing positions; the players change the formation (SS level) in order to be better adapted to the opponent.

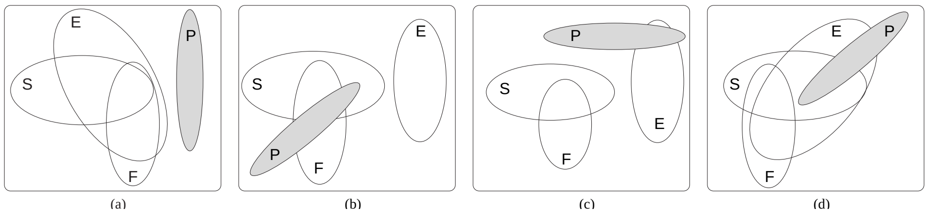
While we can identify two kinds of *changing objects*, we can also identify some types of *changing processes*:

1. Predefined changes: the reorganization is already planed. For example, a soccer team has previously accorded to change its formation at the 30 minutes of the match [13].
2. Controlled (*top-down*): the system does not know when it will reorganize, but when the reorganization is necessary, it will be carried out by a known process (e.g. a diagnostic system controls the reorganization [5]). This process might be performed in two ways: (i) an *endogenous* approach where the system's agent (centralized, e.g. [14,12]) or agents (decentralized, e.g. [7,13]) will carry out the process; or (ii) an *exogenous* approach: the MAS user will control the process [11,15].
3. Emergent (*bottom-up*): there is not any kind of explicit control on the reorganization. The reorganization is performed by some agent according to its own methods.

In the case of controlled reorganization, usually the process which changes the organization has the following general phases [12]: Monitoring, design, selection, and implementation. These phases are detailed in the sequel.

The Monitoring Phase. The monitoring phase identifies a situation where the current organization does not satisfy the needs of the MAS: the organization does not collaborates to the achievement of the social purpose. In other words, the current organization constrains the agents' behaviors to those which do not fit the behaviors that draw the social purpose (in the Fig. 2, the characterization of some of these situations are depicted). Such situations may happen, for instance, when the environment has changed [13], the MAS purpose has changed, the performance requirements are not satisfied, the agents are not capable of well playing their roles, a new task request arrives and the current organization is not appropriate [14,12], some organizational fault was detected [5], etc.

The main problem in this phase is *how to identify whether the social purpose is not being achieved because the current organization does not allow it*. Many other reasons may cause the unaccomplishment of the MAS purpose (e.g. the social purpose is impossible to be achieved, $P = \emptyset$). In some cases to change the organization is not helpful (e.g. situations (a) and (b) of the Fig. 2). Even in the case we know the problem can be solved by the reorganization process, the new problem is to identify *which part*



In the situation (a), the purpose behaviors are not allowed neither by the environment nor by the organization. However to change the organization (represented here by the sets S and F) is not helpful since the best it can do is to lead to the situation (b) — because we do not consider changing the set E. In (b), the P's behaviors are allowed by the organization, but the environment does not allow them. In (c), it is possible to achieve the social propose in the current environment, but the organization does not allow it; thus the reorganization process can solve this problem. In (d), the social purpose can be achieved in the current configuration, but the functional specification does not collaborate to it; again the reorganization process can solve the problem.

Fig. 2. Characterization of some organizational fails

of the organization is causing the problem in order to set the correct reorganization level: OE level (the number of agents, the roles they are playing, . . .) or OS level (e.g. situations (c) and (d) of the Fig. 2)).

The Design Phase. Once a modification need is identified by the monitoring, this phase intends to develop a set of possible alternatives for the current organization. The design of this set of alternatives (i) can be based on a search in a library of predefined organizations or (ii) created on demand.

In the first case, the problem is to identify which predefined organization is appropriate for the problem caught by the monitoring phase. For example, this decision may be based on the characteristics of a new task (e.g. [13]) or a new environment (e.g. the opponent team in [13]).

The second case has to deal with yet another problem. Since we may consider the problem of finding out a new organization as a search problem, the hugeness of this search space forces the definition of some heuristics and specialized tools. To solve this problem and design new organizations on demand, many tools are being used: diagnostic expert systems [5], MAS learning [16], case based reasoning [11], etc.

The Selection and Implementation Phase. This phase selects one of the alternatives generated by the previous phase. The main problem is the definition of the criteria to evaluate which proposal is more promising. Normally, the works on reorganization do not clearly identify this phase, since they join it with the design phase.

The problem in the implementation phase is how to change the current running organization without causing many drawbacks. For example, how an agent will deal with the fact that the role it is playing was removed in the new organization? What it will do with the commitments adopted under this extinguished role? Until we know, there is no work addressing these problems.

4 Reorganization upon $\mathcal{M}$OISE$^+$

The reorganization process proposed here does not solve all the problems presented in the previous section. However it attempts to be an open proposal for the reorganization process with the following constraints: i) an OC organization type is assumed, which is, in our case, the $\mathcal{M}$OISE$^+$; ii) only OS level reorganization is considered (nevertheless many properties of this proposal can be applied on the OE level reorganization); iii) the reorganizational phases are performed in an endogenous and decentralized approach.

In this paper, we will place ourself in a top-down approach of the reorganization. As it is one cooperative process, among others, in a MAS, we may thus describe it by an OC specification support given by $\mathcal{M}$OISE$^+$ itself. Following this trend, it will be defined a group and a social scheme where the reorganization are performed.

4.1 Reorganization group

The reorganization process is performed by a group created from the `ReorgGr` specification defined in the Fig. 3. The *soc* role is the root of the role hierarchy, thus every role defined in a $\mathcal{M}$OISE$^+$ organization is a specialization of *soc* and inherit its properties. The *OrgManager* authority on the *soc* means therefore an authority on every role. The *Monitored* is an abstract role[1] which is specialized by roles, expressed in the application organization itself, whose agents will be monitored by a *Monitor*. In other words, all agents that will be monitored must play a *Monitored* sub-role. The *Designer* contains the common properties for designers (*ReorgExpert* and *OrgParticipant*). The *Reorg* is also an abstract role which allows us to easily distinguish the *OrgManager* from the other roles in this group. Thus we can state, for example, that the *Reorg* and therefore all its sub-roles has permission to communicate with the *OrgManager* role.

The general description of the not abstract roles and their structural position follows (a detailed FS of these roles is given in the next section):

1. ***OrgManager***: the only, and only one, agent that will play this role, will be in charge of managing the reorganization process. Thus it has authority on the *soc* agents[2] and so on all agents. The agent playing this role ought to know the current state of the MAS's organization (OS and OE) and has the permission to change it.
2. ***Historian***: the agent that plays this role maintains the history of the organization — a kind of useful information for the monitoring and design phases. Every change either in the OE (role adoption, commitment with missions, goal achievement, etc.) or in the OS (role creation, link creation, change in the cardinalities, etc.) is registered by this agent. The *Historian* will ask the *OrgManager* to inform him all changes it has executed, these events are called *social events*. The agent which adopts this role could be the same that adopts the *OrgManager* role, since they are compatible.

[1] Abstract roles have only a specification purpose, no agent can play them.
[2] This abbreviation must be understood as agents playing the *soc* role.

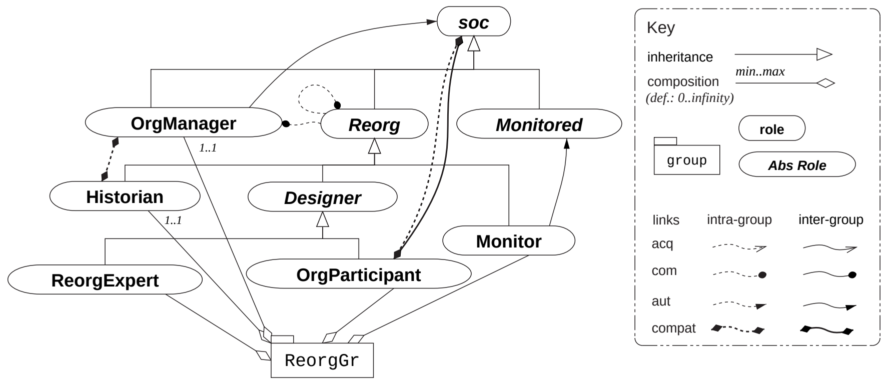


Fig. 3. The reorganization group

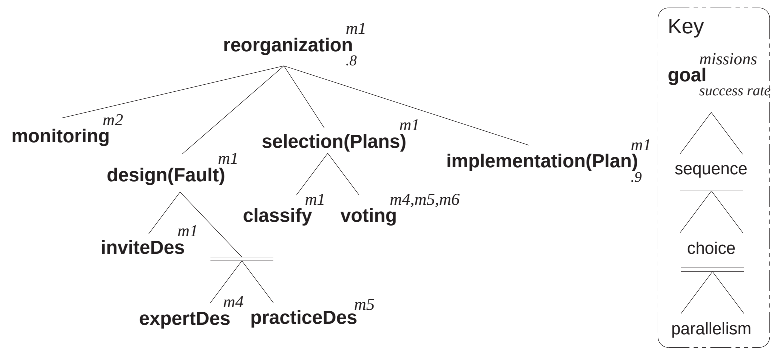


Fig. 4. The reorganization scheme

3. ***Monitor***: agents playing this role will monitor the organization and identify a situation where the reorganization must be performed. The inherited communication link to the *Historian* and the authority on the *Monitored* can be used for its aim. For example, a *Monitor* may look for a predefined metrics like "a SCH's success rate is bellow some threshold" (this Monitor is like a sentinel proposed in [8]).
4. ***ReorgExpert***: agents playing this role have the ability (and the obligation) to analyze the current organization, identify its problems, and propose alternatives. These agents are not allowed to participate in other groups of the MAS since their role are not compatible with any other role. They are invited to participate to the `ReorgGr` just for the reorganization process as a kind of outside analysts which are able to see the organization from a global point of view.
5. ***OrgParticipant***: every agent that plays a role in the MAS is also permitted to play this role, since *OrgParticipant* is compatible with the *soc* role. These agents have practical knowledge about the way the organization works. Conversely to the

ReorgExpert, they are inside analysts and see the organization from a local point of view.

4.2 Reorganization scheme

The agents that have instantiated the `ReorgGr` will perform the reorganization as defined in the SCH of the Fig. 4 and Tab. 1. This SCH is controlled by the *OrgManager* agent which has the obligation for the SCH's root goal and has four main sub-goals (monitoring, design, selection, and implementation) that have to be achieved in sequence.

role	relation type	mission = {goals}
OrgManager	$obligation$	$m_1 = \{$*reorganization, design, inviteDes, selection, classify, implementation*$\}$
Monitor	$obligation$	$m_2 = \{$*monitoring*$\}$
ReorgExpert	$obligation$	$m_4 = \{$*expertDes,voting*$\}$
OrgParticipant	$permission$	$m_5 = \{$*practiceDes,voting*$\}$
OrgParticipant	$permission$	$m_6 = \{$*voting*$\}$

Table 1. Deontic relation for the reorganization SCH

Monitoring Phase. The method that *Monitor* agents will use to achieve their *monitoring* goal (in the mission m_2) is a domain dependent matter. Nevertheless, the $\mathcal{M}$OISE$^+$ may help this phase since the organization description comprises the following useful information for monitoring:

1. the social purpose is explicitly defined and can be verified by some monitor,
2. the SCHs are defined by goals which can also be checked,
3. the global plans have a success rate,
4. the well formed status of the structure can be checked,
5. it is possible to define roles like *Historian* and *Monitored* — and the power these roles have/give — which are useful to collect information for the monitoring.

Once one *Monitor* has decided that a reorganization is required, the *monitoring* goal holds and the next goal (*design*) is allowed. The *Monitor* must send a message to the *OrgManager* telling him the problem that has been identified. This problem description will replace the $Fault$ argument of the *design* goal.

Design Phase. In order to achieve the m_1's *design* goal, the *OrgManager* will firstly invite some agents to play the *Designer* roles (its m_1's *inviteDes* goal). The agents which accept the *ReorgExpert* role ought to commit to the mission m_4 and therefore try to achieve the m_4's *expertDes* goal (design a new organization by expertise) and m_4's *voting* goal (see Tab. 1). Conversely, the agents which accept the *OrgParticipant* role are permitted (not obligated) to commit to the missions m_5 or m_6. In case the *OrgParticipant* commits to the mission m_5, it ought to try to achieve the goals *practiceDes* (design new organization by experimental knowledge) and *voting*. In case this agent commits just to the mission m_6, it only has to collaborate in the achievement of the goal *voting*.

As already stated in the Sec. 3, *Designer* agents may use many methods and tools to achieve their goals. In the `ReorgGr`, each method can be implemented as an agent and the *OrgManager* can invite as many *Designers* as it thinks is enough. In other words, the proposed approach is **open**: as many agents can play the *Designer* role, many tools (eventually very different) can be used in the reorganization process. Rather than stating how the *Designers* will make their *modification proposal*, this group states the social conditions for participating in the reorganization process.

When a *Designer* has developed one modification proposal, it has to write a change plan and to send it to the *OrgManager*. The change plan is formed by actions like role ρ added, role ρ removed, mission m added, obligation added, group specification added, etc. The modification proposals also have one of the following *focus* (the part of the current OS the plan intends to modify): all the current OS, a specific group or role belonging to the SS, a specific scheme or mission belonging to the FS, or relation in the DS.

The Selection and Implementation Phases. As in the two previous phases, the selection is also domain dependent. Nevertheless, we can conceive some selection strategies which may be used in several domains. One possible strategy, based on an early work [10], is an one round voting system where the voter strength depends on both its experience in the society (how many roles does it is currently playing and how many roles did he play) and on the success of its modification proposals (how long have its proposed organizations been active). This strategy also takes into account the cost of the change plan (how many agents will lose their roles or missions in case the proposal is implemented). Once the agents have selected one change plan, the *OrgManager* will perform this plan in order to reorganize the system.

5 Conclusions

This paper has presented a general view and classification of the reorganization problem under the $\mathcal{M}$OISE$^+$ point of view. It is also proposed a reorganization process where the agents have autonomy to change their organizations. However it is done in an OC point of view throughout the specification of a dedicated group. The reorganization group proposed here has been validated in a business to business application that, due to lack of place, we didn't describe here. Our future work will be a validation of this group in a RoboCup team formation.

The $\mathcal{M}$OISE$^+$ organizational model has been shown as a good support for the specification of a MAS's organization which intends to reorganize itself because (i), as an organizational description, it gives useful information for the monitoring and design phases and (ii), as a specification tool, it allows us to define the reorganization process with valuable properties: (a) the openness for many types of monitoring and design; (b) the definition of special roles like the *OrgManager* and *Monitored*; and (c) the specification of the reorganization through the $\mathcal{M}$OISE$^+$ enable any $\mathcal{M}$OISE$^+$ agent to understand and participate in the reorganization.

References

1. Cristiano Castelfranchi. Modeling social action for AI agents. *Artificial Intelligence*, (103):157–182, 1998.
2. J. Galbraith. *Organization Design*. Addison-Wesley, 1977.
3. Les Gasser. Organizations in multi-agent systems. In *Pre-Proceeding of the 10th European Worshop on Modeling Autonomous Agents in a Multi-Agent World (MAAMAW'2001)*, Annecy, 2001.
4. Norbert Glaser and Philippe Morignot. The reorganization of societies of autonomous agents. In Magnus Boman and Walter Van de Velde, editors, *Multi-Agent Rationality*, LNAI 1237, pages 98–111. Springer, 1997.
5. Bryan Horling, Brett Benyo, and Victor Lesser. Using self-diagnosis to adapt organizational structures. In *Proceedings of the 5th International Conference on Autonomous Agentes (Agents' 01)*, 2001.
6. Jomi Fred Hübner, Jaime Simão Sichman, and Olivier Boissier. $\mathcal{M}\text{OISE}^{+}$: Towards a structural, functional, and deontic model for MAS organization. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'2002)*, 2002.
7. Toru Ishida, Les Gasser, and Makoto Yokoo. Organization self-design of distributed production systems. *IEEE Transactions on Knowledge and Data Engineering*, 4(2):123–134, april 1992.
8. Mark Klein and Chrysanthos Dellarocas. Exception handling in agent systems. In *Proceedings of the Third International Conference on Autonomous Agentes (Agent's 99)*, pages 62–68. ACM, 1999.
9. Christian Lemaître and Cora B. Excelente. Multi-agent organization approach. In Francisco J. Garijo and Christian Lemaître, editors, *Proceedings of II Iberoamerican Workshop on DAI and MAS*, Toledo, Spain, 1998.
10. Gustavo Giménez Lugo, Jomi Fred Hübner, and Jaime Simão Sichman. Representação e evolução de esquemas sociais em sistemas multi-agentes: Um enfoque funcional. In Ana Tereza Martins and Díbio Leandro Borges, editors, *Anais do III Encontro Nacional de Inteligência Artificial*, pages 1237–1246, Fortaleza, Brazil, 2001. SBC.
11. Thomas W. Malone. Tools for inventing organizations: Toward a handbook of organizational process. *Management Science*, 45(3):425–443, March 1999.
12. Young-pa So and Edmund H. Durfee. An organizational self-design model for organizational change. In *AAAI93 Workshop on AI and Theories of Groups and Oranizations*, 1993.
13. Peter Stone and Manuela M. Veloso. Task decomposition and dynamic role assignment for real-time strategic teamwork. In Jörg P. Müller, Munindar P. Singh, and Anand S. Rao, editors, *Proceedings of the Fifth International Workshop Agent Theories, Architectures, and Languages (ATAL-98)*, LNCS 1555, pages 293–308. Springer, 1999.
14. Emmanuelle Le Strugeon, Gauthier Agimont, René Mandiau, and Patick Millot. Organisation évolutive d'un système multi-agents: Illustration par des agents-robots miniers. In *Distribuée et Système Multi-Agents*, pages 167–176, Paris. Hermès.
15. Milind Tambe, David V. Pynadath, and Nicolas Chauvat. Building dynamic agent organizations in cyberspace. *IEEE Internet Computing*, 4(2), 2001.
16. Gerhard Weiß. Some studies in distributed machine learning and organizational design. Technical Report FKI-189-94, Institut für Informatik, Technische Universität München, 1994.
17. Michael Wooldridge, Nicholas R. Jennings, and david Kinny. A methodology for agent-oriented analysis and design. In *Proceedings of the Third International Conference on Autonomous Agentes (Agent's 99)*. ACM, 1999.