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

Jomi Fred Hübner *
jomi.hubner@poli.usp.br

Jaime Simão Sichman †
jaime.sichman@poli.usp.br

LTI / EP / USP
Av. Prof. Luciano Gualberto, 158, trav. 3
05508-900 São Paulo, SP

Abstract

The work reported here proposes an Organization-Centered model for controlling the reorganization process of a MAS where the agents themselves try to maintain and adapt the organization to both environmental changes and their purposes. The reorganization process is expressed as a four phases process: 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). The proposed reorganization scheme is evaluated in the robot soccer domain.

1 Introduction

In an organizational centered (OC) point of view [7], 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.

Being well organized is a valuable property of a MAS, since it helps the system to assure its efficacy and efficiency [4]. 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

*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.

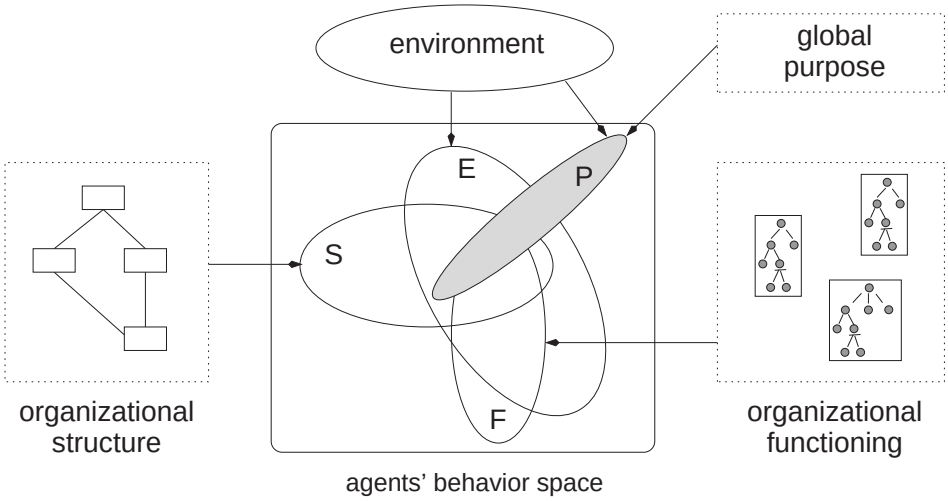


Figure 1: The organization effects on a MAS

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 to little environmental changes 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.

Identifying a good size for the set of organizational allowed behaviors is indeed another way of conceiving one important MAS problem: how to conciliate collective constraints with the agent autonomy . Normally MAS methodologies are concerned with this problem. However, even if the MAS has an initial good organization, dynamic changes either in the environment or global purposes may cause the looseness of the organization suitability. 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 [8].

If we assume that (i) there is no better organization for a context and (ii) different organizations will give different performances for a system [4], 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 reorganization process (Sec. 2.2) and its specification (Sec. 3) based on the $\mathcal{M}$OISE$^+$ model (Sec. 2.1). We will thus show how the reorganization itself could be expressed and controlled in an OC point of view. Before comparing this proposition to related works (Sec. 5), we give a short description of a case study related to soccer robot simulation (Sec. 4).

2 Reorganization within $\mathcal{M}$oise$^+$

Reorganization is a complex process which is dependant of the organizational model which is used. Thus before presenting the different dimensions of the reorganization process that we will consider in this work, let's describe the main features of the $\mathcal{M}$OISE$^+$ model.

2.1 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 an explicit deontic relation among these first two dimensions to better explain how a MAS's organization collaborates for the social purpose and make the agents able to reason on the fulfillment of their obligations or not [6]. 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 entering and/or leaving the OE, group creation, role adoption, mission commitment, etc.

The $\mathcal{M}$OISE$^+$ 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 [3], 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 schemes* 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 enterprize 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 [7] where the first two dimensions can be specified almost *independently* of each other and after properly linked by the deontic dimension.

2.2 Dimensions of reorganization

As we saw in the previous section, in the $\mathcal{M}$OISE$^+$, the organization is built upon two core concepts: (*i*) the static description of the organization (types of roles, groups, links, global plans, etc.), called OS, and (*ii*) the current state of one instance of this OS, called OE, formed by agents with a common social purpose. The reorganization is therefore 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 agent groups, etc.

For example, in a soccer team, the coach can reorganize the team by changing the agent playing the goalkeeper role.

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

- SS level: changes in the groups, roles, and links specification,
- FS level: changes in the Social Scheme (SCH)s, missions, and preferences, and
- DS level: changes in the obligation and permission anchors of missions to roles.

For example, the reorganization can change the team formation (SS level), the team rehearsed plays (FS level), or the players' roles obligations (DS level).

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 and is expressed, for example, as a temporal organization model [2].

   For example, a soccer team has previously accorded to change its formation at the 30 minutes of the match.

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. the team has an expert system that controls the reorganization). This process might be performed in two ways: (*i*) an *endogenous* approach where the system's agent (centralized) or agents (decentralized) will carry out the process; or (*ii*) an *exogenous* approach: the MAS user will control the reorganization process.

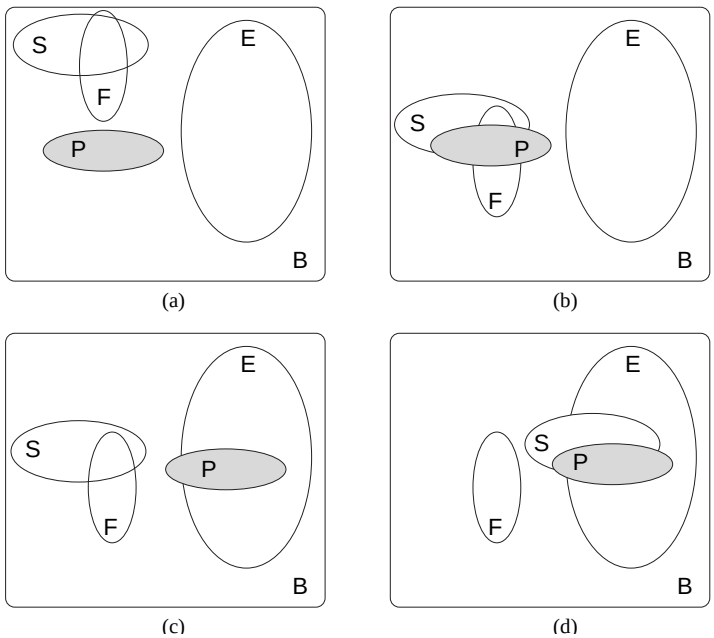
3. Emergent (*bottom-up*): there is not any kind of explicit control on the reorganization. The reorganization is performed by some agent according to its own methods.

Since we are concerned with a controlled reorganization, the reorganization process is composed of the four phases: monitoring, design, selection, and implementation [8]. The problems inherent of these phases are detailed hereafter in sequence.

2.2.1 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 help 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, 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, 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 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).



Given that a well organized system is characterized by $E \cap S \cap F \cap P \neq \emptyset$ and it is not considered changing either P (the purpose) or E (the environment), this figure depicts four fail situations. In situation (a), the purpose behaviors are not allowed neither by the environment nor by the organization. In (b), the P's behaviors are allowed by the organization, but the environment does not allow them; the reorganization does not solve this fail. 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.

Figure 2: Characterization of some organizational fails

2.2.2 The Design Phase

Once a modification need is identified during the monitoring phase, the next step 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 environment, like an opponent team. 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.

2.2.3 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, most research work on reorganization does not clearly identify this phase, since this phase is joined 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? As far as we know, there is no current work in progress addressing these problems.

As we see, the reorganization process is a complex and multi-faceted problem. Moreover, each application domain has its own set of problems leading to different technical solutions for the reorganization phases (case based reasoning, learning, negotiation, etc). In the next section we present an organization model that could express the logic of the reorganization process and constrains the agents participating to the reorganization to follow this logic. The application technical solutions are set in the agents, not in the organization model.

3 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*) a $\mathcal{M}$OISE$^+$ organization type is assumed; *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.

As the reorganization 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 process is performed.

3.1 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

[1]Abstract roles have only a specification purpose, no agent can play them.

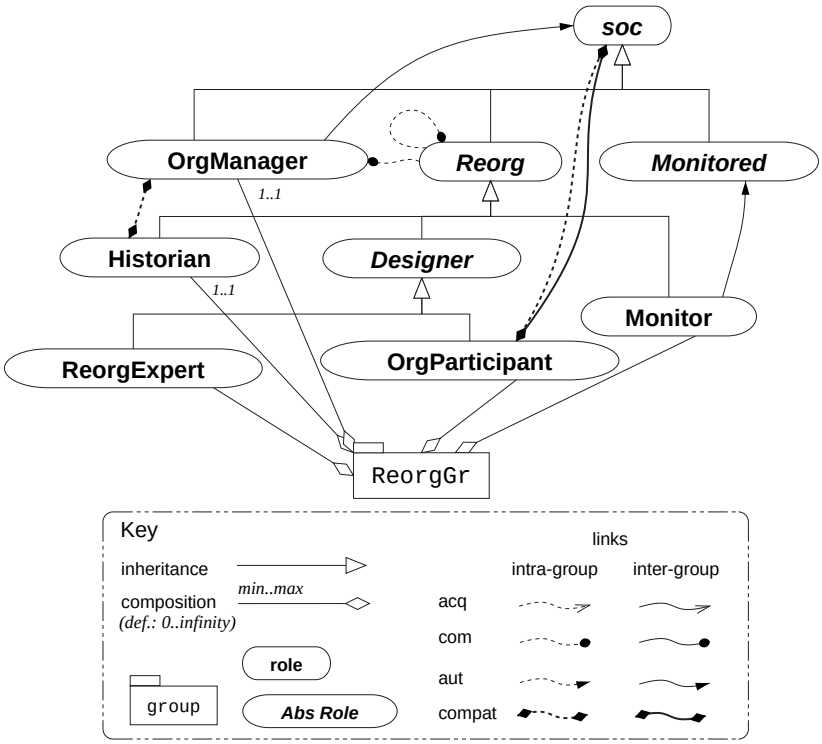


Figure 3: The reorganization group

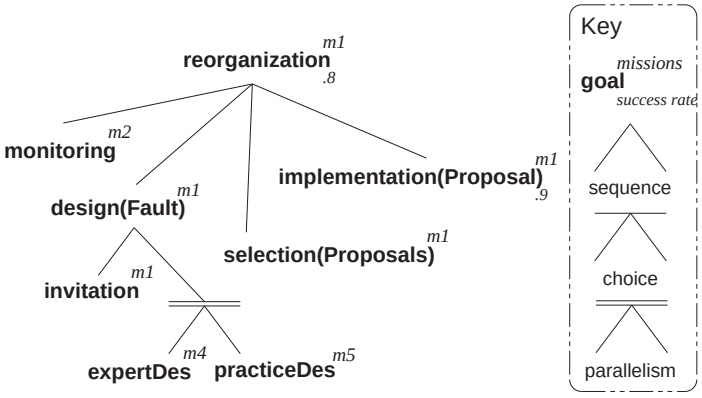
other words, all agents that will be monitored must play a *Monitored* sub-role. The *Designer* contains the common properties for designers (*ReorgExpert* and *OrgParticipant*). The *Reorg* is also an abstract role which allows us to easily distinguish the *OrgManager* from the other roles in this group. 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).

OrgManager: the only, and only one, agent that plays this role, is to be in charge of managing the reorganization process. Thus it has authority on the *soc* agents 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.

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. The agent which adopts this role could be the same that adopts the *OrgManager* role, since they are compatible.

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* help them to achieve their goals.



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

Figure 4: The reorganization scheme

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.

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.

3.2 Reorganization scheme

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

3.2.1 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: the social purpose is explicitly defined and can be verified by some monitor, the SCHs are defined by goals which can also be checked, the global plans have a success rate, the well formed status of the structure can be checked, and 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.

3.2.2 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). Conversely, the agents which accept the *OrgParticipant* role are permitted (not obligated) to commit to the mission m_5. 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).

As already stated in the Sec. 2.2, *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 concept of change plan has two main advantages. Firstly, it defines step by step how the OS will be changed. Thus, when a *Designer* proposes a modification plan it also has to deal with implementations issues like "add the role x and after remove the role y, or remove the role y and after add the role x?". The second advantage is the possibility of change only some part of the OS (the change plan focus), for instance the *Designer* may change the schemes without changing the roles.

3.2.3 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 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). Once the agents have selected one change plan, the *OrgManager* will perform this plan in order to reorganize the system. In the next section, another selection strategy is suggested.

4 Case Study

In order to evaluate the implementability of the proposed reorganization specification, we have done some reorganization experiments on a small size robot soccer league using the TeamBots simulator

[1]. A robot team that follows a $\mathcal{M}$OISE$^+$ specification was developed (it was initially specified in [6]). Roughly, the SS of this team is formed by the players possible roles (goalkeeper, back, attacker, ...) and its field area. The FS is formed by schemes that describe how to score a soccer goal. The agents missions is a set of goals associated to motor schemas that defines the robots behaviors. The team environment is composed by the game score, the game time, and the opponent This team starts each game with a predefined OS and, during the game, is able to change its OS in order to better fit the environment. Due to space limitations, we will describe the reorganization of this team with focus on the reorganization selection phase.

The monitoring phase is simple: the team will reorganize itself each 24,000 simulation steps. Since a game has 120,000 steps, we have 5 reorganizations each game.

The design phase is performed by 8 designer agents playing the *ReorgExpert* role. For instance, one designer always proposes a plan to change the current OS to a new OS where the players area is increased; other designer also focus on the SS and proposes to change the team formation to 1x1x3 (1 back, 1 middle field, and 3 attackers); another designer chooses to focus on the FS and proposes to change the players goals; etc.

The problem is therefore to find out the best sequence of reorganizations that lead the team to win, for instance "in the begin, select the proposal of the 1x1x3 designer, after select the proposal of the designer that use to decrease the area of the goalkeeper, ..., and, near the end of the game, if we are winning, select the proposal of the 1x4 designer". Since this problem can be seen as a Markov Decision Process (MDP) where the environment transition model is unknown, we can use the Q-Learning algorithm to find out the decision policy [10], i.e, in each reorganization which designer proposal to select. After learning this policy, it can be used in the selection phase to choose the reorganization plan with maximum expected reward.

For the Q-Learning specification, the state is the game score and the reorganization time (first, second, ..., fifth reorganization). The opponent, the TeamBots package best team, is fixed, so it is not included in the state representation. The actions set is composed by (a) the action of selecting the proposal of designer i $(1 \leq i \leq 8)$ and (b) not change the organization. The reward of choosing a designer proposal is the number of goals the team scored-suffered. The Fig. 5 shows the team performance when using Q-Learning to learn to select the designer proposals. It takes about 200 games to learn a good policy.

5 Related work

Lots of work has been done on reorganization in MAS. Some, as in [11], use an exogenous approach where the user itself reorganize the whole system. Other, like our proposal, use an endogenous approach where the agent themselves modify the organization. To our knowledge, none of these approaches make clear and explicit the organization controlling the reorganization process itself. The reorganization process is usually hard coded in the MAS itself.

For example, the proposal of [5], a centralized reorganization process with change focus on the FS (described by TÆMS), uses and diagnostic expert system to identify organizational fails and to propose a solution. Its monitoring phase identifies those fails when the system does not behave as expected by its functional model. Our proposal does not have a specific monitoring approach and thus we can have a MAS that *explores* new organizations even in cases no organizational fails occurs (Sec. 4).

The proposal of [9] has a more flexible monitoring phase. Any agent, a soccer player, can identify in the environment the opportunity for reorganization. The reorganization is composed by a change in the team formation (a SS reorganization level in the $\mathcal{M}$OISE$^+$ term) and in the current plan (FS level). Our proposal, besides the explicit organizational model, enable us to consider other modification

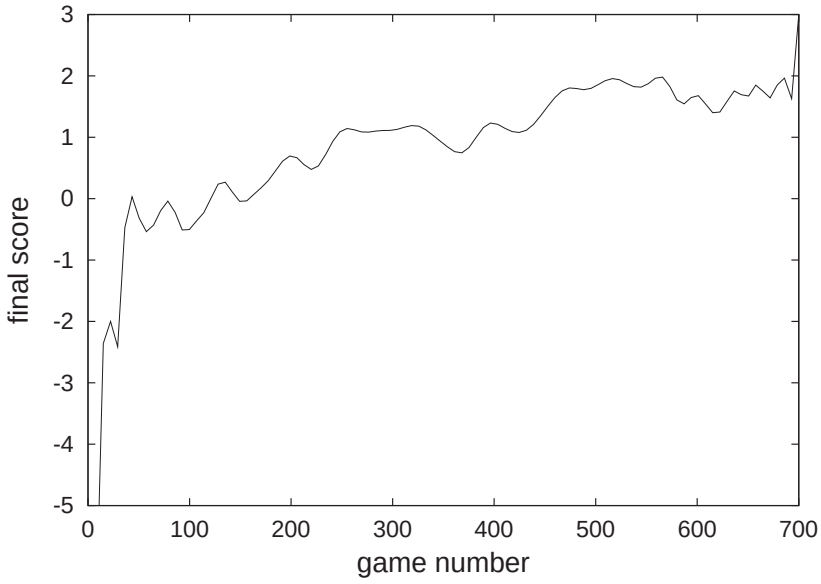


Figure 5: Team performance using Q-Learning to learn to select reorganization proposals

objects, the DS for instance (one can maintain the same roles and change only their obligations to plans).

6 Conclusions

This paper has presented a general view 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. This process is based on an OC point of view throughout the specification of a dedicated reorganization group.

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] Tucker Balch. Teambots, 2000. `http://www.teambots.org`.

[2] Thibault Carron and Olivier Boissier. Towards a temporal organizational structure language for dynamic multi-agent systems. In *Pre-Proceeding of the 10th European Workshop on Modeling Autonomous Agents in a Multi-Agent World (MAAMAW'2001)*, 2001.

[3] Cristiano Castelfranchi. Modeling social action for AI agents. *Artificial Intelligence*, (103):157–182, 1998.

[4] 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.

[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. A model for the structural, functional, and deontic specification of organizations in multiagent systems. In Guilherme Bittencourt and Geber L. Ramalho, editors, *Proceedings of the 16th Brazilian Symposium on Artificial Intelligence (SBIA'02)*, LNAI 2507, pages 118–128, Berlin, 2002. Springer. `http://www.inf.furb.br/~jomi/pubs/2002/Hubner-sbia2002.pdf`.

[7] 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*, 1998.

[8] Young-pa So and Edmund H. Durfee. An organizational self-design model for organizational change. In *Proceedings of AAAI93 Workshop on AI and Theories of Groups and Organizations*, 1993.

[9] 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 5th International Workshop Agent Theories, Architectures, and Languages (ATAL-98)*, LNCS 1555, pages 293–308, Berlin, 1999. Springer.

[10] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Bradford, Cambridge, 1998.

[11] Milind Tambe, David V. Pynadath, and Nicolas Chauvat. Building dynamic agent organizations in cyberspace. *IEEE Internet Computing*, 4(2), 2001.