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

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

[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. Reorganisation within a multi-agent system may be managed by the agents themselves by adapting the organisation to both environmental changes and their own goals. In this paper, we propose an organisation-centred model for controlling this process. Using the $\mathcal{M}$OISE$^+$ organisation model, we are able to define an organisational structure bearing on a reorganisation process along four phases: monitoring (when to reorganise), design (ways of building a new organisation), selection (how to choose an organisation), and implementation (how to change the current running organisation). The proposed reorganisation scheme is evaluated in the robot soccer domain where we have developed players that follow the team organisation specified in $\mathcal{M}$OISE$^+$. A special group of agents may change this organisation, and thus the team behaviour, using reinforcement learning for the selection phase.
Keywords: Autonomous Agents and Multi-Agent Systems, MAS organizations, groups, and societies; reoranization.

1 Introduction

The organisation 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 [3]. The Fig. 1 briefly shows how an organisation could explain or constrain the agents' behaviour in case we consider an organisation as having both *structural* and *functional* dimensions. In this figure, it is supposed that an MAS has the purpose of maintaining its behaviour in the set P, where P represents all behaviours which draw the MAS's social purposes. In the same figure, the set E represents all possible behaviours in the current environment. The MAS's organisational structure is formed, for example, by roles, groups, and links that constrain the agents' behaviour to those inside the set S, so the set of possible behaviours ($E \cap S$) becomes closer to P. It is a matter of the agents, and not of the organisation, to conduct their behaviours 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 behaviours active.

Being well organised is a valuable property of an MAS, since it helps the system to assure its efficacy and efficiency [5]. Our general view of the organisation for an MAS, depicted in the Fig. 1, allows us to state a minimal condition for an MAS to be well organised: $E \cap S \cap F \cap P \neq \emptyset$, i.e., the behaviours which lead to the social purpose achievement are allowed by the organisation in the current environment. However it is almost impossible (indeed undesirable) to specify an organisation where the allowed agents' behaviours fit exactly the set P, since this set also depends on the environment. Different environments require different sets of P behaviours. 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 organisation. On the other side, if S and F are too big, the organisation will not be effective since the agent's behaviours are not sufficiently constrained.

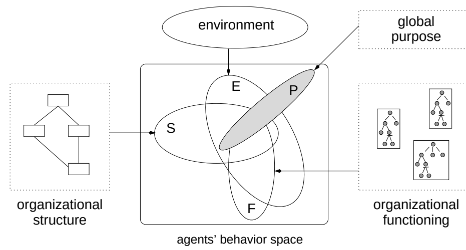


Fig. 1. The organization effects on a MAS

Identifying a good size for the set of organisational allowed behaviours 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 in the MAS design phase (e.g. [14]). However, even if the MAS has an initial good organisation, dynamic changes either in the environment or in the global purposes may cause the looseness of the organisation suitability. Moreover, if we consider the organisation unchangeable, the agents which have several experience and information about the organisation can not contribute to its adaptation. They loose the autonomy regarding its organisation. In other words, this problem could be expressed as how to conciliate an agent centered (AC) point of view with an organizational centered (OC) point of view [8]. This situation brings the *reorganisation* problem up: how the agents themselves might change their current organisation [10].

If we assume that (i) there is no better organisation for a context [4] and (ii) different organisations will give different performances for a system [5], an MAS needs to be capable of reorganising itself in order to well suit in its environment and to efficiently achieve its goals. Our objective is therefore to propose a reorganisation model and its specification (Sec. 3) based on the $\mathcal{M}$OISE$^+$ (Sec. 2). We will thus show how the reorganisation 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 Reorganisation within $\mathcal{M}$OISE$^+$

The $\mathcal{M}$OISE$^+$ (Model of Organisation for multI-agent SystEms) follows the general view of the organisation depicted in the Fig. 1 and therefore considers the organisational structure and functioning. However, this model adds an explicit deontic relation among these first two dimensions to better explain how an MAS's organisation collaborates for the social purpose and make the agents able to reason on the fulfilment of their obligations or not [7]. These three dimensions form the Organisational Specification (OS). When a set of agents adopts an OS they form an Organisational 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 reorganisation is therefore a process which changes the current state of the OS or OE 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).

While we can identify two kinds of *changing objects* (OS or OE), we can also identify some types of *changing processes*:

1. Predefined changes: the reorganisation is already planed and is expressed, for example, as a temporal organisation 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 reorganise, but when the reorganisation is necessary, it will be carried out by a known process (e.g. the team has an expert system that controls the reorganisation). This process might be performed in two ways: (i) an *endogenous* approach where the system's agent (centralised) or agents (decentralised) will carry out the reorganisation; or (ii) an *exogenous* approach: the MAS user will control the reorganisation process.
3. Emergent (*bottom-up*): there is not any kind of explicit control on the reorganisation. The reorganisation is performed by some agent according to its own methods.

Since we are concerned with a controlled reorganisation, the reorganisation process is composed by four phases: monitoring, design, selection, and implementation [10]. The problems inherent of these phases are detailed hereafter in sequence.

The Monitoring Phase. The monitoring phase identifies a situation where the current organisation does not satisfy the needs of the MAS. In other words, the current organisation constrains the agents' behaviours to those which do not fit the behaviours that draw the social purpose. 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 organisation is not appropriate, etc.

In the Fig. 2, the characterisation of some of these situations are depicted. Given that a well organised system is characterised 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 the Fig. 2 situation (a), the purpose behaviours are not allowed neither by the environment nor by the organisation. In (b), the P's behaviours are allowed by the organisation, but the environment does not allow them; the reorganisation does not solve this two first fails. In (c), it is possible to achieve the social propose in the current environment, but the organisation does not allow it; thus the reorganisation 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 reorganisation process can solve the problem.

The main problem in this phase is *how to identify whether the social purpose is not being achieved because the current organisation 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 organisation is not helpful (e.g. situations (a) and (b) of the Fig. 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 organisation.

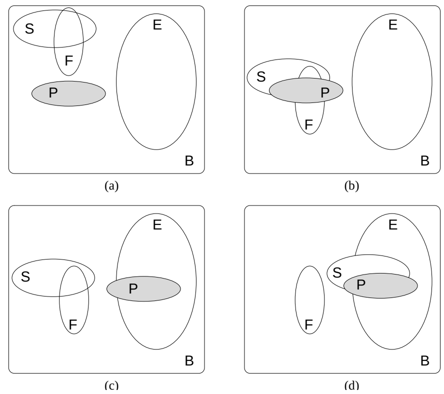


Fig. 2. Some organisational fails

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*. The problem in the implementation phase is how to change the current running organisation 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 organisation? 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 briefly see, the reorganisation 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 reorganisation phases (case based reasoning, learning, negotiation, etc). In the next section we present a reorganisation model that could express the logic of the reorganisation process and constrains the agents participating to the reorganisation to follow this logic.

3 Reorganisation upon $\mathcal{M}$OISE$^+$

The reorganisation model proposed here does not solve all the problems presented in the previous section. However it attempts to be an *open* proposal for the reorganisation process with the following assumptions: i) a $\mathcal{M}$OISE$^+$ organisation type is assumed; ii) only reorganisation at the specification level is considered (nevertheless many properties of this proposal can be applied on the entity level reorganisation); iii) the reorganisational phases are performed in an endogenous and decentralised approach.

As we conceive the reorganisation as one cooperative process among others in an MAS, we may thus describe it by the specification support given by $\mathcal{M}$OISE$^+$ itself. Following this trend, the next sections describe a group and a social scheme where the reorganisation process is performed.

Reorganisation group. The reorganisation process is performed by a set of agents that play roles inside a group created from the group specification defined in the Fig. 3, this group is identified by `ReorgGr` (the graphical notation of the $\mathcal{M}$OISE^{+} specification language is not detailed here, the reader is referred to [7] for more information). The *soc* role is the root of the role hierarchy, thus every role defined in a $\mathcal{M}$OISE^{+} organisation inherits its properties.

The agent that assumes the *OrgManager* role is to be in charge of managing the reorganisation process, it is able to change the current state of the MAS's organisation (OS and OE). It also has authority on the *soc* agents and so on all agents.

Monitored is an abstract role[1] which is specialised by roles defined in the application organisation. All agents that will be monitored must play a *Monitored* sub-role and thus are under the *Monitor* agent authority since the *Monitor* role has authority on the *Monitored* role.

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*.

The *Historian* agent maintains the history of the organisation — 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.

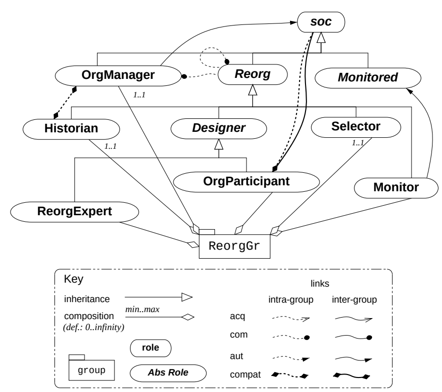


Fig. 3. The reorganisation group

The *Designer* role contains the common properties for designers. Agents playing *ReorgExpert* role have the ability (and the obligation) to analyse the current organisation, identify its problems, and propose alternatives. They are invited to participate to the `ReorgGr` just for the reorganisation process as a kind of outside analysts which are able to see the organisation from a global point of view. Conversely, every agent that plays a role in the MAS is also permitted to play the *OrgParticipant* role, since *OrgParticipant* is compatible with the *soc* role. These agents have practical knowledge about the way the organisation works. They are inside analysts and see the organisation from a local point of view.

Finally, the agent that plays the *Selector* role is responsible for the selection of one proposal from the reorganisation proposals developed by the *Designer* agents.

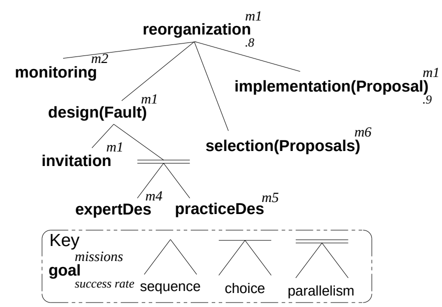
The set of agents that will play these roles is called *reorgConf*. While some of the reorgConf agents must be developed for each specific domain (such as the monitor, selector, and designers), some of them can be used in many applications (such as the

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

OrgManager and the Historian). All of them must follow the behaviour constraints defined by the `ReorgGr` group and the reorganisation scheme defined in the next section.

Reorganisation scheme. The reorgConf that has instantiated the `ReorgGr` will perform the reorganisation as defined in the scheme shown in the Fig. 4. This scheme is controlled by the *OrgManager* agent which has the obligation for the scheme's root goal. The reorganisation scheme decomposes the root goal in four sub-goals (monitoring, design, selection, and implementation) that have to be achieved in sequence by the agents compromised with the scheme's missions ($m_1, m_2, ..., m_6$). In the $\mathcal{M}$OISE$^+$, when an agent assumes a role (*ReorgExpert*, for instance) and some scheme is created (the reorganisation scheme, for instance), this agent has obligation or permission for this scheme's missions as stated in the deontic specification (the table of the Fig. 4, for instance).

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 organisation description comprises the following useful information for monitoring: the social purpose is explicitly defined and can be verified by some monitor, the schemes 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.



role	relation type	mission = {goals}
OrgManager	$obligation$	m_1 = {*reorganisation, design, invitation, implementation*}
Monitor	$obligation$	m_2 = {*monitoring*}
ReorgExpert	$obligation$	m_4 = {*expertDes*}
OrgParticipant	$permission$	m_5 = {*practiceDes*}
Selector	$obligation$	m_6 = {*selection*}

Fig. 4. The reorganisation scheme

Once one *Monitor* has decided that a reorganisation 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 organisation 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 organisation by experimental knowledge).

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 reorganisation process. Rather than stating how the *Designers* will make their proposals, this group states the social conditions for participating in the reorganisation process.

In order to achieve its goal, a *Designer* has to write a *plan of changes* and send it to the *OrgManager*. The plan of changes have to modify, step by step, the current organisation to a new organisation. It is formed by actions like add/remove a role, a mission, an obligation, or a group specification. The plan of changes 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 Structural Specification (SS), a specific scheme or mission belonging to the Functional Specification (FS), or relations in the Deontic Specification (DS).

The concept of plan of changes has two main advantages. Firstly, it defines step by step how the OS will be changed. Thus, when a *Designer* proposes a plan of changes 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 plan of changes focus), for instance the *Designer* may change the schemes without changing the roles.

The Selection and Implementation Phases. As in the two previous phases, the selection is also domain dependent. In the next section, a selection strategy is suggested. Once the *Selector* agent has selected one plan of changes, the *OrgManager* will perform this plan in order to reorganise the system.

Although implementation issues are not covered in this paper, the implementation of the reorgConf agents is helped by the $\mathcal{M}$OISE$^+$ architecture, available at `http://www.lti.pcs.usp.br/moise`. This architecture is composed by some general propose agents (as OrgManager), a communication infrastructure, and an agent architecture that follows the $\mathcal{M}$OISE$^+$ organisational specification and can be extend for specific applications. However, it is important to note that the agents must *follow* the organisation constraints and not be implemented based on it (as suggested by some MAS methodologies) since the organisation may change during the agent's life.

4 Case Study

In order to evaluate the implementability of the proposal, we have done some reorganisation 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. The agent architecture is based on a multi-layer approach. The top layer is the *organisational layer* which links the agent to the organisation. It enables the agent (a) to know the OS (groups, roles, schemes, ...); (b) to produce organisational events like group creation, role adoption, and mission commitment; and (c) to know its obligations and permission regarding position on the organisation. The *deliberative layer* is responsible for choosing a role and a mission for the agent. Having a mission and therefore its goals, this layer set the motor schema of the reactive layer that achieves the selected goal. The *reactive layer* perceives the environment and reacts to it according to its current motor schema [9].

Roughly, the initial organisational structure of this team is formed by five possible roles and their field area (Fig. 5). The team also has a reorganisation sub-group as defined in Sec. 3. 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' behaviour [9]. The team environment is composed by the match score, the match time, and the opponent. This team starts each match with a predefined OS and, during the match, is able to change its OS in order to better fit to the environment.

The reorgConf of the team is composed by one monitor, nine designers, and one selector. The monitor agent is very simple, it starts a reorganisation each 24,000 simulation steps. Since a match has 120,000 steps, we have 5 reorganisations each match. This monitoring strategy is justified by the exploration property of the Reinforcement Learning (RL) algorithm used by the selector agent.

The design phase is performed by 9 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.

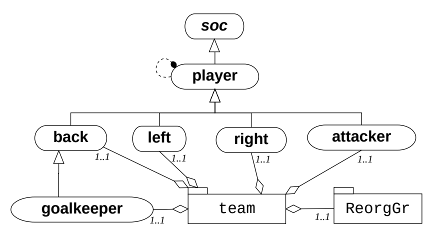


Fig. 5. Example of the team structure

The problem is therefore to find out the best sequence of reorganisations 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 [12], i.e, in each reorganisation which designer proposal must be selected. After learning this policy, it can be used in the selection phase to choose the reorganisation plan with maximum expected reward.

For the Q-Learning specification, a state s is a pair formed by the game score and the reorganisation time (first, second, ..., fifth reorganisation). The opponent, the TeamBots package best team, is fixed, so it is not included in the state representation. The actions set A is composed by the action of selecting the proposal of designer i ($1 \leq i \leq 9$) and not change the organisation. The reward r of choosing a designer proposal is the number of goals the team scored minus the number of suffered goals after this proposal has been implemented. At each reorganisation, the selector agent updates the Q-values by the following rule: $Q(s,a) \leftarrow Q(s,a) + \alpha \left(r + \gamma \max_{a' \in A} Q(s',a') - Q(s,a)\right)$ where s and a represent the last reorganisation state and action, α is the learning rate (the initial value is 0.2 and decays 0.0001 each match), γ is the future discount rate (we use 0.9), and s' is the next state in case the action a' is performed (see [12] for RL algorithm's parameters). Based on the current state s", the selector then chooses the next action by the following function

$$\epsilon - greedy(s") = \begin{cases} \text{ramdom action from } A & \text{if } rv < \epsilon \\ \arg\max_a Q(s", a) & \text{otherwise} \end{cases}$$

rv is a random value ($0 < rv \leq 1$) and ϵ ($0 \leq \epsilon \leq 1$) is the exploration rate (it starts with 0.5 and decays 0.0001 each match).

The Fig. 6 shows the team final score along 2000 matches when using Q-Learning to learn to select the designer proposals. It takes about 480 matches to learn a good selection policy, i.e., a good sequence of reorganisations during a match. Thus, with this particular reorgConf, we have an MAS that learns to reorganise itself according to its environment (the opponent). The selection and implementation problems presented at the Sec. 2 are solved in our proposal by the RL and the concept of plan of changes. Of course, this case study does not aim at the soccer problems, but it has exemplified how the proposed reorganisational model could be successfully applied.

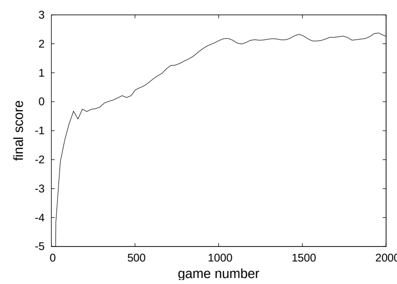


Fig. 6. Learning results

5 Related work

Lots of work has been done on reorganisation in MAS. Some, as in [13], use an exogenous approach where the user itself reorganise the whole system. Other, like our proposal, use an endogenous approach where the agent themselves modify the organisation. To our knowledge, none of these approaches make clear and explicit the organisation controlling the reorganisation process itself. The reorganisation process is usually hard coded in the MAS itself.

For example, the proposal of [6], a centralised reorganisation process with change focus on the FS (described by TÆMS), uses a diagnostic expert system to identify organisational 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 an MAS that *explores* new organisations even in cases no organisational fails occurs (Sec. 4).

The proposal of [11] has a more flexible monitoring phase. Any agent, a soccer player, can identify in the environment the opportunity for reorganisation. The reorganisation is composed by a change in the team formation (a structural reorganisation in $\mathcal{M}$OISE$^+$ terms) and in the current plan (functional level). Our proposal, besides the explicit organisational model, enable us to consider another modification objects, the deontic specification for instance (our proposal can maintain the same roles and change only their obligations to plans).

6 Conclusions

This paper has presented a general view of the reorganisation problem under the $\mathcal{M}$OISE$^+$ point of view. The main contribution is a reorganisation model where the agents have autonomy to change their organisations. This process is based on an OC point of view throughout the specification of a dedicated reorganisation group.

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

References

1. T. Balch. Teambots, 2000. `http://www.teambots.org`.
2. T. Carron and O. Boissier. Towards a temporal organizational structure language for dynamic multi-agent systems. In *Proceedings of the MAAMAW'2001*, Annecy, 2001.
3. V. Dignum and F. Dignum. Modelling agent societies: Co-ordination frameworks and institutions. In P. Brazdil and A. Jorge, editors, *Proceedings of the EPIA'01*, pages 191–204, Berlin, 2001. Springer.
4. J. Galbraith. *Organization Design*. Addison-Wesley, 1977.
5. L. Gasser. Organizations in multi-agent systems. In *Pre-Proceeding of the MAAMAW'2001*, Annecy, 2001.
6. B. Horling, B. Benyo, and V. Lesser. Using self-diagnosis to adapt organizational structures. In *Proceedings of the Agents'01*, 2001.
7. J. F. Hübner, J. S. Sichman, and O. Boissier. A model for the structural, functional, and deontic specification of organizations in multiagent systems. In G. Bittencourt and G. L. Ramalho, editors, *Proceedings of the SBIA'02*, pages 118–128, Berlin, 2002. Springer.
8. C. Lemaître and C. B. Excelente. Multi-agent organization approach. In F. J. Garijo and C. Lemaître, editors, *Proceedings of II Iberoamerican Workshop on DAI and MAS*, 1998.
9. R. R. Murphy. *Introduction to AI robotics*. Bradford Book/MIT Press, 2000.
10. Y. So and E. H. Durfee. An organizational self-design model for organizational change. In *Proceedings of AAAI93 Workshop on AI and Theories of Groups and Organizations*, 1993.
11. P. Stone and M. M. Veloso. Task decomposition and dynamic role assignment for real-time strategic teamwork. In J. P. Müller, M. P. Singh, and A. S. Rao, editors, *Proceedings of the ATAL-98*, pages 293–308, Berlin, 1999. Springer.
12. R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. Bradford, Cambridge, 1998.
13. M. Tambe, D. V. Pynadath, and N. Chauvat. Building dynamic agent organizations in cyberspace. *IEEE Internet Computing*, 4(2), 2001.
14. M. Wooldridge, N. R. Jennings, and D. Kinny. A methodology for agent-oriented analysis and design. In *Proceedings of the Agent's 99*. ACM, 1999.