Chapter 39
Distribution of Roles in Virtual Organization of Agents

Juan F. De Paz, Carolina Zato, Gabriel Villarubia, Javier Bajo and Juan M. Corchado

Abstract Agent-based virtual organizations have acquired growing relevance during the last years. While these systems can be used to model human societies, there are still some open issues to be solved when working with agent-based virtual organizations, including conflict resolution. In a virtual organization it is possible to find different conflictive situations among which is the task of assigning roles to the agents in the organization. The number of agents can vary dynamically in an organization, which has produced the need to define automatic self-adaptive mechanisms for role assignment. This paper presents an innovative linear programming mechanism for role assignment in virtual organizations of agents. With the use of linear programming it is possible to determine the roles that will be assigned to each agent based on the agent's specific capabilities. The proposed mechanism was tested in a case study in geriatric residences and the results obtained are presented in this paper.

Keywords Virtual organizations · Lineal programming · Distribution of roles

J. F. De Paz (✉) · C. Zato · G. Villarubia · J. M. Corchado
Department of Computer Science and Automation,
University of Salamanca Plaza de la Merced,
s/n 37008 Salamanca, Spain
e-mail: fcofds@usal.es

C. Zato
e-mail: carol_zato@usal.es

J. M. Corchado
e-mail: corchado@usal.es

J. Bajo
Departamento de Inteligencia Artificial. Facultad de Informática,
Universidad Politécnica de Madrid,
Madrid, Spain
e-mail: jbajo@fi.upm.es

L. Uden et al. (eds.), *The 8th International Conference on Knowledge Management in Organizations*, Springer Proceedings in Complexity,
DOI: 10.1007/978-94-007-7287-8_39,

39.1 Introduction

Nowadays, the inclusion of organizational aspects is achieving more importance in the research of multi-agent systems. The evolution of multi-agent systems to virtual organizations of agents introduced concepts derived from social organizations, such as, roles, groups and norms. One of the first attempts to include organizational concepts in multi-agent systems occurs in [1]. As one of the earlier systems, MACE, introduced the concept of role to describe multi-agent systems [2]. Other examples of early works on the use of organizational concepts in MAS come from [3] and [4] but they do not tackle the problem of how to design such organizations. Work on organizational theories started within the context of the CommonKADS effort, but has since focused on human organizations models [5].

An exception is found in [6] which describes an analysis method for MAS in organizational terms.

Once the concept of organization and role was established, the next challenge was the organization itself in such structures. The agent concept has evolved from an autonomous entity that behaves individually, almost selfishly, to a collaborative entity that is part of a society, group or organization. Thus, interaction, coordination and adaptation are key points of an agent society and are closely related to organization and reorganization, which can determine behaviors and, clearly, communications channels.

The organization of agents can be considered from different perspectives, although the most common is by grouping organizations with common interests. An interest can be the desire or need to share resources and competencies, or to solve a common goal. Comparing artificial agents to those from the real-world, an agent can have different roles and be part of various organizations. The reorganization of an agent organization can be motivated by the desire to reduce conflicts within inter-agent cooperation and to increase the efficiency in achieving goals [1]. The process of reorganization must take roles into account since a role reflects the competencies that an agent either has or should provide to other agents [7].

The real world is highly dynamic and quick, which is why computer applications should be adaptable and able to follow automatic and efficient methods. Multi-Agent Systems (MAS) are often cited as one of the most promising approaches to create open systems. However, these open MAS themselves and their environments are not static; they change, disappear or grow. Agents can migrate, organizational objectives can change, and operational behavior can evolve [8]. Continuous change requires continuous reorganization and/or restructuring, and since the VO paradigm is based on human societies, it is logical that it should also perform efficient reorganizations. The problem is when and how. A high dynamicity involves a high need of reorganization, which in turn makes it necessary to find an effective, automatic and formal way to order the societies.

In this paper, an organization is considered solely on the basis of its structure, i.e. by the way groups and roles are arranged to form a whole, without being concerned with the way agents actually behave. Additionally, multi-agent systems will be analyzed from the "outside", as a set of interaction modes [7]. Therefore, no issues regarding the architecture of agents or the way agents act will be addressed. Agents will only be defined by their functions in an organization, i.e. by their roles, and by the sets of norms, which they must follow to play those roles.

This paper presents a role distribution model for agents of an organization. The model is based on minimizing a certain parameter. The model uses linear programming to search for the optimal solution. This ensures that the best solution can be found when we are working with linear restrictions and objective function. The system has been integrated into the PANGEA [9] architecture, allowing a distribution of roles among the available agents by following a structure of organization, constraints, capabilities and costs of the agents. The system was validated in a case study to verify the correct operation of the proposal.

This article is divided as follows: Sect. 39.2 describes the state of the art; Sect. 39.3 presents the proposed model; Sect. 39.4 describes the results obtained and the conclusions respectively.

39.2 Background

An organizational structure can be defined as the set of group structures expressing the design of a multi-agent organization scheme [7]. The principles of reorganization are clearly explained in [8]. The most popular implementations of organizational adaptation include approaches based on load balancing [10, 11] or dynamic task allocation [12, 13]. The latter is often the case in organizational self-design in emergent systems that, for example, include composition and decomposition primitives that allow for dynamic variation of the organizational structure (macro-architecture) while the system population (micro-architecture) remains the same [14]. Another common approach is dynamic participation, where agent interaction with the organization is modelled with roles, and adaptation occurs as agents move in and out of those roles [15–17]. However, few of these systems allow agents to change the problem-solving framework of the system itself [18]. Based on the above considerations, two reorganization situations can be identified: the behavioural change related to roles; and the structural change related to interaction between agents, dependencies and norms.

This study is interested in how organizations change, and how reorganization can occur dynamically, with minimal interference from the system designer. From our point of view behavioural changes are strongly closed to structural changes.

One concern is the starting point in generating an organization. To create an organization it is necessary to have an efficient distribution of roles among the available agents. The topologies represented in Fig. 39.1 illustrate different types of organization and communication that can exist for agents, including [19]:

- Hierarchies: agents are arranged in a tree-like structure in which the lower levels have basic functionalities and higher levels have decision making and control.

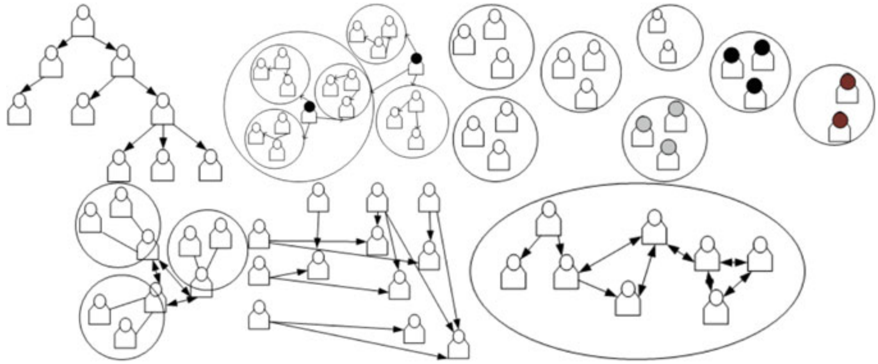

Fig. 39.1 *Top* row, *left* to *right*: hierarchy, holarchy, coalition, congregation, *Bottom* row, *left* to *right*: federation, matrix organization y group

- Holarchies are nested and hierarchical structures of holons [20]. A holon is a part of larger entity, and is the result of an association of subordinated entities. This type of topology is usually applied in domains where the objectives are recursively broken down into subtasks.
- Coalitions are temporal associations of agents that are created to achieve a particular goal, often generating benefits and reducing costs. Coalitions are dissolved once the objective is achieved. Internally they are usually represented as a flat structure or with a leader, while externally they are represented as a single and atomic entity.
- Groups are sets of cooperative agents working together towards a common goal. Thus, they maximize the usefulness and efficiency of the equipment. The representation of goals, beliefs and plans is made at a team level. The groups are usually created when solving problems can be best achieved by working together. While the groups lend themselves to greater redundancy, they are more flexible. However, greater flexibility leads to more communication and increased difficulty to coordinate.
- Congregations are groups of agents with similar or complementary characteristics. In this case, they do not work to achieve a specific goal, instead they facilitate the process of finding suitable partners to achieve that goal. For this reason, this type of topology is usually intended for long-term goals.
- Federations are groups of agents with a representative. The other members of the organization interact only with the representative, giving up part of their autonomy. This agent "representative" also acts as an intermediary between the group and the outside world, taking on various functions such as (i) Broker: distributes tasks among group members (ii) Mediator: facilitates interactions between different actors (makes contacts) (iii) Monitor: controls states of agents and reports on events (iv) Embassy: controls communication between external agents and agents of the federation (translator).

- Matrix Organizations: in this type of topology organization, an agent can be controlled by more than one agent supervisor. For this reason, it is necessary to use mechanisms for evaluating commitments and local conflict resolution. It is like a grid-like structure in which the manager agents surround other agents.

The presented method allows reorganization of any topology and can be easily adapted to personalized topologies.

39.3 Proposed Reasoning System

The reorganization mechanism has been included as a behaviour inside the agent that plays the role of OrganizationAgent in the PANGEA platform. PANGEA [9] is a multi-agent platform to develop open multiagent systems, specifically those including organizational aspects such as virtual agent organizations. This platform allows the integral management of organizations and offers different tools to the end user. Additionally, it includes a communication protocol based on the IRC standard, which facilitates communication and remains robust even with a large number of connections.

PANGEA provides the following pre-determined roles that must be fulfilled by at least one agent to ensure the proper functioning:

- OrganizationManager: responsible for the actual management of organizations and suborganizations. It is responsible for verifying the entry and exit of agents, and for assigning roles when an agent enters the organization for first time. To carry out these tasks, it works with the OrganizationAgent, which is a specialized version of this agent.
- OrganizationAgent: works closely with the OrganizationManager in charge of performing organization issues.
- InformationAgent: responsible for accessing the database containing all pertinent system information.
- ServiceAgent: responsible for recording and controlling the operation of services offered by the agents.
- NormAgent: ensures compliance with all the refined norms in the organization.
- CommunicationAgent: responsible for controlling communication among agents, and for recording the interaction between agents and organizations.
- Sniffer: manages the message history and filters information by controlling communication initiated by queries.

39.3.1 OrganizationAgent

The creation process of the virtual organizations and the distribution of roles among the agents may be carried out in a distributed or centralized way. This process becomes more complicated when an initial organization is not available and the system needs to assign an initial distribution of roles among the agents.

This distribution can be carried out using agreement technologies, allowing the agents to exchange messages and negotiate the distribution of roles. This process can take place when the virtual organization requires small modifications. However, it would become very difficult to establish an initial distribution or create a new distribution when a lot of changes are necessary, as this situation would require many messages. An innovative solution to solve this problem consists of applying linear programming when the first distribution of roles is created, or when there are many changes with regard to the number of agents or the roles that need to be assigned.

For each of the organization types, we define a different graphical representation. For example, Fig. 39.2a represents a virtual organization with a hierarchical structure. The organizational structure is stored in an XML file similar to the one shown in Fig. 39.2b. The system has to fill out the identifiers of the agents responsible for the roles. A specific role is assigned to one agent, although one agent may have several roles and the same role may be assigned to different agents.

In addition to the information associated with the required structure, it is necessary to define the information associated with each of the agents, taking into account their ability to perform certain roles. According to these characteristics and to a series of restrictions, it is possible to establish a linear programming problem to assign each tag agent to an agent identifier. The information stored for each of the agents varies according to the case study taken into consideration.

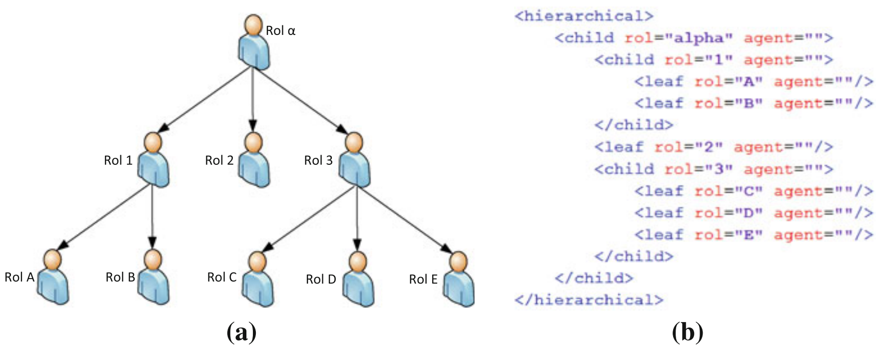

Fig. 39.2 **a** Graphical representation of the organization. **b** XML with the information of the organization

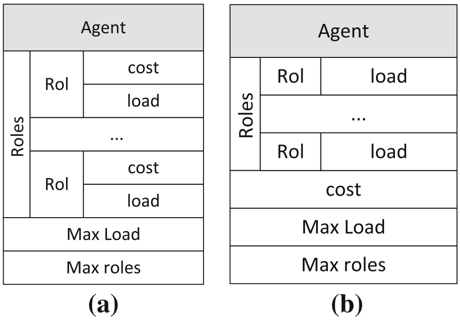

Fig. 39.3 **a** Restrictions with costs per roles. **b** restrictions with costs per blocks

An example of restriction for the organizational structure shown in Fig. 39.2 can be seen in Fig. 39.3a. In addition to the information stating the ability of an agent to play a specific role, and the corresponding work load, we have included information which will apply restrictions for the maximum number of roles that an agent can handle at one time. Figure 39.3a contains information about the cost and load of carrying out a role, in addition to some variables that limit the maximum load of the agent and the number of assigned roles. Figure 39.3b is similar to 39.3a although it additionally contains a cost value associated to the maximum load. For example a block of time can represent a working day.

Once the information of the structure and the restrictions is available, the optimization problem can be resolved. In this case, a minimization problem is formulated. The next subsection formulates the integer linear programing problem used to resolve the optimization problem.

39.3.1.1 Optimization Problem

To explain the problem simply, it is necessary to create a network containing all the Information about abilities and maximum load for the agents. The assignation problems represent the information in a graph in order to facilitate the formulation of the optimization problem. The graph contains values that are situated over the connecting lines and contain information about the cost and capacity of carrying out a task. Based on this information and the values it is easy to represent the optimization problem. Figure 39.4a shows the information previously presented in Fig. 39.3a. The roles are labeled with the name r_i where i represents the role. The agents are labeled with the name a_i. The nodes 0 and f are incorporated to establish the start and end points. The values v_{ij} establish the cost of the agent j to carry out role i. The values c_{ij} represent the load of role i over the agent j. The values l_i indicate the max load of the agent i. Figure 39.4b shows the restrictions according to Fig. 39.3b.

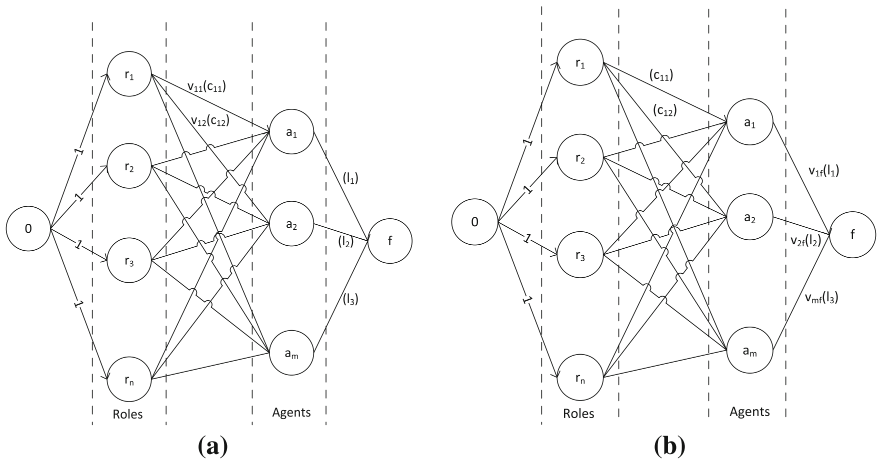

Fig. 39.4 **a** Representation of the information and restrictions according to the cost for each role. **b** representation of the information and restrictions according to the cost for max load

The optimization problem represented in Fig. 39.4a is resolved according to the optimization problem formulated in the following lines. The Branch and Bound and simplex formula [21] is used to resolve the optimization problem. The variable x_{ij} represents a connection between role i and agent j, the other variables are defined at the beginning of this section.

$$
\begin{aligned}
&Min\, v_{11}x_{11} + v_{12}x_{12} + \ldots + v_{nm}x_{nm} \\
&\qquad st \\
&c_{11}x_{11} + \ldots + c_{n1}x_{n1} \le l_1 \\
&\qquad \ldots \\
&c_{1m}x_{1m} + \ldots + c_{nm}x_{nm} \le l_m \\
&x_{11} + \ldots + x_{1m} \ge 1 \\
&\qquad \ldots \\
&x_{n1} + \ldots + x_{nm} \ge 1 \\
&0 \le x_{ij} \le 1 \\
&x_{ij}\ hold\, number
\end{aligned}
$$

The optimization problem associated with the Fig. 39.4b is different, and will be revised in the case study presented in Sect. 39.4.

39.4 Results and Conclusions

The system was applied to a previously analyzed case study to test the performance of role distribution among the medical personnel of a geriatric residence. In this residence, the personnel needed for a working day is composed of the following specializations (roles): Three attendants, 15 nurses, three floor supervisors, 15 caregivers, and one manager. The geriatric residence has a total of 50 employees (agents) that can be used to carry out the working day tasks. The proposed system has to assign the roles to these individuals. Each employee is capable of carrying out several tasks according to the abilities required for the task, the load of each task, and the max load of the person. Table 39.1 shows the information available about each individual and the roles previously identified. That information is then used to assign the roles. As an example, Table 39.1 shows one role per specialty and five agents that have been represented. However, all personnel are taken into account and represented in the table. The information in the table follows the model represented in Fig. 39.3b. The maximum number of roles per agent has not been limited because it is indirectly limited by the max load parameter. The information from Table 39.1 is represented as follows: cost of the working day | necessary time to carry out the role | time of the working day | maximum number of roles. The cost is a function of the working day; that is, if an agent carries out two roles in the working day, the price will be the same as if the agent carries out only one role. As can be seen in Table 39.1, a role can imply a different workload for each of the agents, even if the role requires the same capacities. The loads are assigned according to the experience of each employee in the residence. For example when considering the assignment of role floor supervisor, it is possible to observe that agent 3 requires 3 h while the agent 2 would need only 2 h to carry out the job. Additionally, as can be seen, agents 2 and 3 receive a different salary. Agent 5 is qualified to act as manager, which requires 2 h. That leaves agent 5 with 6 additional hours to perform any other compatible role.

Using the information presented in the Table 39.1, the optimization problem can be formulated as shown below:

Table 39.1 Table of indexes for each of the agents and each of the roles. The information is represented in the following format: working day cost (v_{ij}) | role load time (c_{ij}) | maximum load of the agent l_j | maximum number of roles

	Agent1	Agent2	Agent3	Agent4	Agent5	…
Order1	80|8|8|∞					…
Nurse1		120|6|8|∞	110|7|8|∞			…
Supervisor1		120|2|8|∞	110|3|8|∞		150|2|8|∞	…
Carer1		120|6|8|∞	110|6|8|∞	100|6|8|∞		…
Manager1					150|2|8|∞	…
…	…	…	…	…	…	…

$$
\begin{aligned}
& Min\, 80x_{1f} + 120x_{2f} + 110x_{3f} + 100x_{4f} + 150x_{5f} \\
& st \\
& 8x_{11} \leq 88 \\
& x_{1f} - x_{11} \geq 0 \\
& 6x_{22} + 2x_{32} + 6x_{42} \leq 8 \\
& x_{2f} - x_{22} \geq 0 \\
& x_{2f} - x_{32} \geq 0 \\
& x_{2f} - x_{42} \geq 0 \\
& 7x_{23} + 3x_{33} + 6x_{43} \leq 8 \\
& x_{3f} - x_{23} \geq 0 \\
& x_{3f} - x_{33} \geq 0 \\
& x_{3f} - x_{43} \geq 0 \\
& 6x_{44} \leq 8 \\
& x_{4f} - x_{44} \geq 0 \\
& 2x_{35} + 2x_{55} \leq 8 \\
& x_{5f} - x_{35} \geq 0 \\
& x_{5f} - x_{55} \geq 0 \\
& x_{11} \geq 1 \\
& x_{22} + x_{23} \geq 1 \\
& x_{32} + x_{33} + x_{35} \geq 1 \\
& x_{42} + x_{43} + x_{44} \geq 1 \\
& x_{55} \geq 1 \\
& x_{11} + x_{22} + x_{32} + x_{42} + x_{23} + x_{33} + x_{43} + x_{44} + x_{35} + x_{55} = 5 \\
& 0 \leq x_{ij} \leq 1 \\
& x_{ij}\, Integer\, number
\end{aligned}
$$

The variable x_{ij}, where *i* and *j* are numbers, represents a connection between role *i* and agent *j*. x_{if} represents a connection between agent i and final node f of the network represented in Fig. 39.4b. The variables x_{ij} and x_{if} are integers and the value may be 0 or 1 according to the restrictions. The other variables are defined above in Sect. 39.3.1.1

The generic assignation problem is defined as follows. The first of the restrictions implies that the max load is not greater than the maximum hours an agent can work, while the second restriction implies that the cost of the working day of the agent is active if a role has been assigned to the agent. For example, in the previous example $x_{2f} - x_{32} \geq 0$ if the role 3 is assigned to agent 2 then $x_{32} = 1$ and then $x_{2f} = 1$. The third restriction implies that each role is assigned to at least one agent, and the last restriction represents that each role is assigned to only one agent.

Table 39.2 Final costs obtained with the different proposals

Algorithm	Test 1	Test 2	Test 3	Test 4	Test 5
Manual	3430	4110	3200	3380	3580
Genetic algorithm	3380	3720	3340	3350	3410
Linear programming	3220	3580	3200	3280	3410

$$
\begin{aligned}
& Min \sum_i v_{if} x_{if} \\
& st \\
& \forall j \quad \sum_i c_{ij} x_{ij} \le l_j \\
& \forall i,j \quad x_{jf} - x_{ij} \ge 0 \\
& \forall i \quad \sum_j b_{ij} x_{ij} \ge 1 \\
& \sum_{i,j} b_{ij} x_{ij} = n \\
& 0 \le x_{ij} \le 1
\end{aligned}
$$

$c_{ij} = 0 y b_{ij} = 0$ when there is no link between role i and agent j, or the cell row i col c_{ij} is empty in Table 39.1. Otherwise value is the cost and $b_{ij} = 1$, where n is the number of roles to assign.

The assignation of the roles and tasks would be defined as the values of the variables x_{ij}. A value of 1 indicates that role i is assigned to agent j, 0 if it is not assigned. The result of the problem shown above is as follows: agent 1 performs the role of attendant, agent 2 has no roles, agent 3 performs the role of nurse, agent 4 is assigned to a caregiver, agent 5 performs the role of floor supervisor and manager, and the cost is 440€.

To evaluate the performance of the system, the total cost of the assignation problem using the 50 workers and the 37 roles was compared to both a manual planner and a generic automatic planner. The manual plan is carried out by a staff member who distributes the roles among the available personal according to the restrictions, while the automatic plan is based on a genetic algorithm. During the test, the number of agents and capacities of the personnel were modified, while the number of the roles remained constant. The results of the test are shown in Table 39.2. As can be seen, the proposed approach reduces the costs in personnel and in all cases the total cost is equal to or lower than the other two approaches.

The planning model of the agent OrganizationAgent in the PANGEA architecture [9] allows the reduction of costs and the simplification and increased efficiency of the process for assigning roles. Figure 39.3a and b represent two different problems with two of the more common situations in this kind of system. These models could be easily extended to other problems if all the restrictions and the objective functions are linear. In other cases, the optimization problem should be resolved with heuristics. The case study was applied to the model presented in Fig. 39.3b. In future works, new organization models and restrictions will be analyzed to incorporate them into the agent OrganizationAgent in the PANGEA architecture.

Acknowledgments This work has been supported by the MICINN TIN 2009-13839-C03-03.

References

1. Glaser N, Morignot P (1997) The reorganization of societies of autonomous agents. Multi-agent rationality, Lecture Notes in Computer Science. Vol 1237, pp 98–111
2. Gasser L, Braganza C, Herman N () Mace a flexible testbed for distributed ai research. In: Huhns MN (ed) Distributed artificial intelligence, pp 119–152
3. Fox M (1981) An organizational view of distributed systems. IEEE Trans Man Syst Cybern 11(1):70–80
4. Prasad MVN, Decker K, Garvey A, Lesser V (1996) Exploring organizational designs with TAEMS: a case study of distributed data processing. In: Tokoro M (ed) Proceedings of second international conference on multi-agent sytems, pp 283–290
5. Hoog R, Benus B, Metselaar C, Vogler M, Menezes W (1994) Organisation model: model definition document. Technical report, ESPRIT Project P 5248
6. Collinot A, Drogoul A (1996) Agent oriented design for a soccer robot team. In: Kyoto MT (ed) Proceedings of second international conference on multi-agent systems. AAAI Press, Kyoto, pp 41–47
7. Ferber J, Gutknecht O (1998) A meta-model for the analysis and design of organizations in multi-agent systems. Proceedings of international conference on multi agent systems, 1998, pp 128–135
8. Dignum V, Dignum F, Sonenberg L (2004) Towards dynamic reorganization of agent societies. In Proceedings of workshop on coordination in emergent agent societies, pp 22–27
9. Zato C, Villarrubia G, Sánchez A, Bajo J, Rodríguez S, Corchado JM (2012) PANGEA—platform for automatic construction of organizations of intelligent agents. Distributed computing and artificial intelligence. Adv Intell Soft Comput 151:229–239
10. Montresor A, Meling H, Babaoğlu Ö (2003) Messor load-balancing through a swarm of autonomous agents. Agents and peer-to-peer computing. Lecture notes in computer science. Vol 2530, pp 125–137
11. Young JL, Geon YP, Ho KS, Hee YY (2012) A load balancing scheme for distributed simulation based on multi-agent system. IEEE 36th annual computer software and applications conference workshops (COMPSACW), pp 613–618
12. Kivelevitch E, Cohen K, Kumar M (2011) Market-based solution to the allocation of tasks to agents. Procedia Comp Sci 6:28–33
13. Palmer D, Kirschenbaum M, Murton J, Zajac K, Kovacina M, Vaidyanathan R (2003) Decentralized cooperative auction for multiple agent task allocation using synchronized random number generators. Proceedings of IEEE/RSJ international conference on intelligent robots and systems (IROS 2003)
14. So Y, Durfee E (1993) An organizational self-design model for organizational change. In: Proceedings of AAAI93 Workshop on AI and Theories of Groups and Organizations, Washington, DC, USA, pp 8–15
15. Dignum V (2004) A model for organizational interaction: based on agents, founded in logic. SIKS Dissertation Series, Utrecht University PhD Thesis
16. Glasser N, Morignot P (1997) The reorganization of societies of autonomous agents. In: MAAMAW, pp 98–111

17. Tambe M (1997) Towards flexible teamwork. J Artif Intell Res 7:83–124
18. Barber KS, Martin CE (2001) Dynamic reorganization of decisionmaking groups. Proceedings of the 5th autonomous agents
19. Rodríguez S (2010) Modelo adaptativo para organizaciones virtuales de agentes. University of Salamanca, Department of Computer Science and Automation, PhD Thesis (2010)
20. Mazigh B, Hilaire V, Koukam A (2011) Formal specification of holonic multi-agent systems: application to distributed maintenance company. In: Proceeding of 9th international conference on practical applications of agents and multiagent systems
21. Zheng Y, Lin MC, Manocha D (2011) Efficient simplex computation for fixture layout design. Comput Aided Des 43(10):1307–1318
22. Zato C, De Paz JF, de Luis A, Bajo J, Corchado JM (2012) Model for assigning roles automatically in egovernment virtual organizations. Expert Syst Appl 39(12):10389–10401