A Multi-agent Organizational Model for Grid Scheduling

Inès Thabet[1,2], Issam Bouslimi[1,2], Chihab Hanachi[1], and Khaled Ghédira[2]

[1] Institut de Recherche en Informatique de Toulouse IRIT, UMR 5505,
Université Toulouse 1 Capitole, 2 rue du Doyen-Gabriel-Marty, 31042 Toulouse Cedex 9
{Ines.Thabet,Issam.Bouslimi,Chihab.Hanachi}@irit.fr
[2] Stratégie d'Optimisation des Informations et de la ConnaissancE,
Université de Tunis, 92, Avenue 9 avril 1938, Tunis - 1007, Tunis-Tunisie
Khaled.Ghedira@isg.rnu.tn

Abstract. Multi-agent technology provides high level organizational concepts (groups, roles, commitments, interaction protocols) to structure, coordinate and ease the adaptation of distributed systems efficiently. This paper proposes to model a grid scheduling system as a multi-agent system organization. The resulting organizational model, based on the Agent Group Role meta-model of Ferber, is evaluated at the conceptual and implementation level. At the conceptual level, we evaluate the efficiency, robustness and flexibility of our model. At the implementation level, the analysis and the evaluation of our proposition, done through simulations, show its efficiency.

1 Introduction

One motivation of Grid computing [1] is to aggregate the power of widely distributed computational resources and thus to offer the possibility to solve very large problems. To achieve this goal, an efficient Grid Scheduling system is required. The *Grid scheduling system* is in charge of allocating users' applications tasks on a set of feasible and adequate resources following an objective function.

Scheduling systems are difficult to build due to several constraints and notably the dynamicity of the environment and the complexity of the interactions among the different components. These interactions are of paramount importance for finding resources, allocating them tasks and for supporting adaptation in case of disturbances.

The *complexity* of the interactions is due to the fact that the scheduling system is made of several *heterogeneous* and *autonomous* components that have to interact frequently and cooperate at a high level to perform efficiently users' applications in a *disturbed* environment. The *heterogeneity* is due to the different roles involved in a scheduling system (grid scheduler, local scheduler and resource) and the huge number of their occurrences with different features and capacities. Besides, these different components are most of the time *autonomous*, since each one has its own management and access policy. The scheduling may also be *perturbed* by the *dynamicity* of the grid. This dynamicity is due to the open character of the grid (resources can join or leave freely) and to performances variability of the resources.

J. O'Shea et al. (Eds.): KES-AMSTA 2011, LNAI 6682, pp. 148–158, 2011.

Giving this context, one of the main issues to face is *how to design, implement and deploy an adaptive grid scheduling system taking into account the previous difficulties*. Moreover, we should deal with this issue at *a macro level* (regulation of the components) since we can't control and act on the internal behavior of each component (micro-level) which are dynamic, huge and above all unknown a priori.

Unfortunately, the examination of the state of the art shows the insufficiency of existing systems to face the previous issue. While agent technology has already been used with success [2] to manage the distribution and the autonomy of the Grid (see the AGEGC system [3], the CONOISE project [4], the AGRD_NFRP system [5]), most of these systems do not take into consideration the dynamicity of the environment. However, we must mention the AppleS agent system [6] that implements an adaptive scheduling. The main drawback of AppleS is that it has to be integrated in the application to be executed and generates a lot of works for the developers. Besides, all these systems impose an internal structure to the agents and do not consider their organization and regulation to reach the system objective.

In order to tackle this issue, we choose to *model the grid scheduling system with an agent organizational perspective* [7] as a means to structure and regulate the scheduling system. This perspective has the advantage of abstracting a system at a high level with macro and social concepts such as roles, interaction protocols, groups, obligations, etc. Not only this organizational model should improve the efficiency of the scheduling system at run time, but it could also be considered as a conceptual tool to capture, model and implement the structure and the behavior of this system in a modular way. Moreover, the concept of agent organization is now mature, and techniques to make it flexible and adaptive are today available [8] and could be used with benefit for the adaptation of the grid scheduling.

More precisely, the proposed organizational model follows the Agent Group Role (AGR) meta-model of Ferber [9]. This model has been designed, implemented and evaluated. Our contributions are threefold:

- *An organizational model* for grid scheduling, which identifies the roles involved in our system, the links that exist between them and the protocols that rule their interactions. This model is general enough to capture any grid scheduling system.
- *Performance criteria* for evaluating our proposed organizational model based on the adaptation of [10] and [11] works. The criteria include structural and statistical aspects. The *structural aspect* is based on the communication network topology. The *statistical aspect* is concerned with the communication and execution time.
- *An implementation and evaluation* of our model. The implementation is based on the Madkit platform [12] that integrates the AGR meta-model. Our evaluation measures the performance criteria mentioned above.

The remainder of this paper is organized as follows. In Section 2, an overview of the proposed organizational model for grid scheduling is presented. We propose performance criteria and the evaluation of our organizational model for grid scheduling in Section 3. Finally, we summarize and lay out the future work.

2 Overview of the Organizational Model for Grid Scheduling

In order to specify and describe the functioning of the grid scheduling system (GS) and more precisely how the agents interact, we choose to model the GS with an organizational view. Introducing organizational abstractions makes it possible to structure and regulate agent's interactions in order to organize their works, limit conflicts and reduce communication cost. For that purpose, we have used the Agent Group Role (AGR) meta model which is appropriate to the Grid context. In AGR, ***agents*** are actives communicating entities that can participate in several ***groups*** (communities) in parallel and can play one or several ***roles*** in each group. No constraint is placed on the internal structure of the agent. AGR has several advantages useful in our context: it eases the *modularity* through the organization of agents' tasks and their interactions in logical entities. Also, AGR allows the design of *open* systems such as grid since new agents could enter or leave the system provided that they could play predefined roles. The system can also be designed with *heterogeneous* components since agent internal architectures are not constraints. Finally, the *reusability* is facilitated since the organizational approach identifies and makes available recurring interaction patterns which become reusable.

The remainder of this section first presents the Grid Scheduling architecture and the proposed organizational model for Grid Scheduling.

2.1 General Architecture of a Grid Scheduling System

Following previous work, we defined a multi-agent based architecture for the grid scheduling system. See [13] for details. In this section we describe the architecture.

The ***User Interface*** allows the interaction between the user and the Global Scheduler (GS) Agents. Using the user interface, the user submits a set of applications made by a set of coordinated tasks, to be executed on the Grid, to the GS.

The ***Global Scheduler Agents*** (GS) are responsible for managing one or several domains, scheduling application's users on a set of adequate resources. A GS orchestrates applications following their processes and selects the best pool of resources in order to optimize its objective function. Moreover, the GS is responsible for monitoring, if necessary adapting the grid functioning in case of disturbances, collecting the execution results and reporting them to the user. The adaptation mechanism will be detailed in the next section and are basically based on the use of organizational structure adaptation, on rescheduling and on interaction protocols.

The ***Local Scheduler Agents*** (LS) are responsible for managing one or several resources belonging to the same domain. A Local Scheduler is also responsible for tasks scheduling in a domain according to the local policy. Moreover, the LS is in charge of detecting failure and for adapting tasks' allocation if possible.

Resources provide high computing capabilities to enable task execution. Each resource is managed by a ***Resource*** **Agent (RA)** in charge of reporting information (resource state and availabilities, tasks execution progress, execution results) to the Local Scheduler.

2.2 An AGR Organizational Model for Grid Scheduling

In this section, we describe our organizational model which identifies the actors implied in our system, the links that exist between them, and the protocols ruling their interactions. Our model is composed of the following components (see Fig. 1):

- Five types of groups represented by an ellipse and described below.
- Three types of agents represented by a candle that are: *Global Scheduler*, *Local Scheduler* and *Resource* Agents. The multiplicity of the agents participating to a group is represented by a star inside the candle.
- Five types of roles since the Local Scheduler and the Resource Agents play two different roles in different groups. The defined roles are: *Global Scheduler (GS)*, *Local Scheduler*, *Resource*, *Resource Application* and *Virtual Organization Scheduler* Roles. Role is represented as a hexagon and a line links this hexagon to agents. The Resource Application Role concerns a dedicated resource that offers one or several services (here the service is the task execution) with a certain quality (dependent of the resource execution capacity). The mapping between the application's tasks types (skill or capacity required by the resource to execute this task) and the resource capacities entails a virtual organization formation made by specialized and distributed physical resources. When a Virtual Organization is formed, an agent playing the role Virtual organization Scheduler is launched by the GS in order to orchestrate the tasks in the virtual organization.

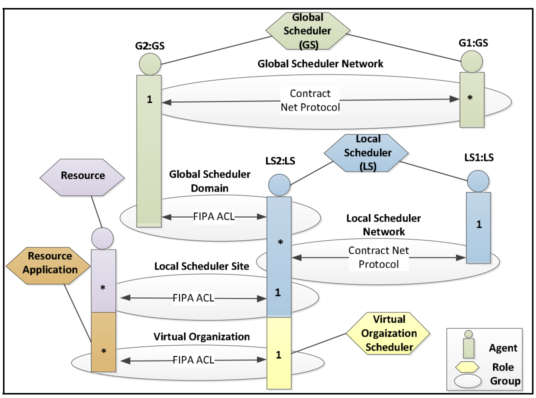

Fig. 1. Organizational Multi-agent based Grid Architecture

The communication between agents in different groups follows the FIPA-ACL language that supports high level communication between the Grid components.

Let us introduce the defined groups and detail how does each group operates.

The ***Global Scheduler (GS) Network*** Group allows the GS to cooperate in order to allocate application's tasks efficiently when resources, under a GS control, are not available or in the case of disturbance. In this context, GS agents cooperate in order to sub-contract tasks' executions using the *Contract Net (CN) Protocol* [14] .The use of the CN is justified since it is one of the most flexible and efficient negotiation mechanisms and it eases tasks distribution. Moreover, it allows the GS managing the contract to choose the better GS bidder regarding its objective. In our current implementation, the GS optimizes the application makespan.

The ***Global Scheduler (GS) Domain*** Group is composed of a GS and its Local Scheduler (LS) agents. This group allows the GS to allocate the application's tasks to the adequate LS, to be executed on its resources, based on the GS objective, resources states and task requirements. Moreover, this group allows the LS to send execution results and resources states, and to inform its GS if problems occur on resources.

The ***Local Scheduler (LS) Site*** Group is composed of the Local Scheduler and the resources at its disposal (in the local site). This group allows a LS to allocate the submitted tasks to the local resources, to collect the resources states (busy, free, broken), the execution progress and results, etc. In the case of resources breakdown, the LS try to reschedule the failed tasks on other resources belonging to its group according to the local resources state. If there is no adequate resource, the LS uses the Local Scheduler Network to find adequate LS acquaintance for tasks execution.

The ***Local Scheduler (LS) Network*** Group allows the LS agents to cooperate efficiently in order to sub-contract tasks execution when resources, at the LS disposal, are not available, busy, broken, etc. The LS agents cooperate using the Contract Net Protocol. The LS initiator evaluates the received offers (execution estimation) and sends tasks to the selected LS or notifies the Global Scheduler about the failure. In this last case, the GS performs an adequate adaptive action.

Let us notice that in the previous groups, adaptation is performed by to the use of high-level protocols, like contract net protocol that eases task allocation even if the Grid environment evolves. The use of interaction protocols allows selecting dynamically resources to face the failure without varying the organization's structure.

The ***Virtual Organization*** (VO) Group is composed of specialized resources, dedicated to a type of task, and by a Local Scheduler playing the role of Virtual Organization Scheduler (VS). The VS is responsible of coordinating the functioning of the Resources Application (dedicated resource) and monitoring their execution. The Virtual Organization formation made by the Global Scheduler depends on the offered services, QoS and resources availabilities. The adaptation in this case consists in reorganizing the virtual organization by removing overloaded resources, adding new better specialized resources, replacing dedicated resources by others, etc. The adaptation results in the modification of the structure of the virtual organization.

3 Evaluation of our Model

3.1 Performance Metrics

In this section we introduce the performance criteria used in our approach and the evaluation of our model for grid scheduling based on the following criteria. These criteria are organized according to two aspects: the *structural* and the *statistical* level:

The ***structural level*** concerns the communication topology of the organization and is based on the Grossi's measures [10] that will be described below. These measures allow to proof organizational qualities such as the *flexibility* (the capacity to adapt easily to changing circumstances), the *robustness* (how stable the organization is in the case of anticipated risks) and the *efficiency* (the amount of resources used to perform its tasks) by using relations between roles. In our case, we will evaluate these qualities and proof that our model is sufficiently robust and efficient.

The ***statistical level*** concerns the performance's evaluation of our organizational model at run time. To this end, two metrics will be observed: the *application execution time* and the *number of messages* in scenarios without perturbations and with perturbations. The comparison of the number of messages with and without perturbations is commonly used for evaluating adaptive multi-agents system [11].

Lets' us details the ***Structural level*** based on Grossi's [10] measures. Evaluating the organizational structure qualities involves three steps:

1. The first step consists in building a role graph of the organization based on the possible relations between two roles. To do that, Grossi introduces three dimensions characterizing these relations:

- The *power structure* defines the task delegation patterns. The existence of power link between agents "*a*" and agent "*b*" means that every delegation of tasks from "*a*" to "*b*" ends in creating an obligation to "*b*".
- The *coordination structure* concerns the flow of information in the organization.
- The *control structure.* An agent "*a*" controls an agent "*b*" if "a" monitors "b"' s activity and "a" possibly takes over the tasks which "b" has not accomplished.

2. Secondly, for *each dimension* (power, coordination, control), a set of concepts and equations from the graph theory are applied in order to measure specific qualities (robustness, flexibility and efficiency) of organizational structures (OS):

- The *connectedness* of an OS shows the connection degree between roles. The more this degree is high, the more the structure can be divided in logical groups.
- The *economy* of an OS expresses how to keep the structure connected while minimizing the number of links between roles (redundant links must be avoided).
- The *univocity* allows us to have an idea on the degree of ambiguity in the OS. For example stating that an agent "a" controls an agent "b" and in the same way, "b" controls "a" generates some ambiguity.

3. Finally, these previous concepts are compared with optimum values defined by Grossi (See [10] for more details) in order to measure the quality of the OS.

3.2 Evaluation of Our Organizational Model

In this section we describe the structural and statistical evaluation of our model.

3.2.1 Structural Evaluation

According to the dimensions described above (power, coordination, control), we generate the role graph (Fig. 2) corresponding to our organizational model. Our graph includes the roles described in the section 2.2. Also, we distinguish between the role "Global Scheduler (GS) Initiator" that is a GS playing the role of tasks manager and the role "Global Scheduler Contractor" corresponding to bidders. The same separation has been made for the Local Scheduler (LS) role.

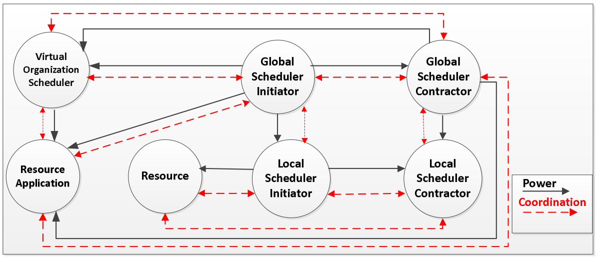

Fig. 2. Role graph

Our graph includes only the power and coordination dimensions represented as edges:

- *Power* expresses the delegation mechanism. In our context, the GS Initiator can delegate tasks to all the other roles except to the resources only managed by LSs (Initiator and Contractor). The Virtual Organization Scheduler (VS) has a power relation with the Resources Application. The LS Initiator also has a power relation with its acquaintances (LS contractor) and with the Resources under its control.
- *Coordination* is a symmetric relation in our graph. It consists in information exchange or protocols-based interaction between roles. It concerns the same couple of roles as the power relation.

The *Control* relation is not present in our organization, since we do not model the evaluation of agents' behavior by their acquaintances. In future work, we intend to refine our model to include this dimension. This assumes to model the continuous monitoring of the agent work to check whether agents are doing their tasks as agreed.

Following this role graph and the set of equations introduced in [10], we compute the three following organizational qualities: the robustness, flexibility and efficiency as shown in the tables below:

Table 1. Structural measures of robustness

Optimum Values	Economy $_{Coord}$	0	Univocity $_{Power}$	0	Connect$_{Coord}$	1
Obtained Values		5/9		1/7		1

Table 2. Structural measures of flexibility

Optimum Values	Connect$_{Power}$	0	Connect$_{Coord}$	1	Economy$_{Coord}$	0
Obtained Values		1		1		5/9

Table 3. Structural measures of efficiency

Optimum Values	Economy $_{Power}$	1	Economy $_{Control}$	1	Economy $_{Coord}$	1
Obtained Values		8/9		1		5/9

Comparing our results to the optimum ones, we can state that our system is sufficiently *robust* and *efficient* but not enough *flexible*. Lets' us explain these results:

Robustness of an organization requires a coordination structure highly connected and weakly economic and a power structure weakly univocal (see Table1). In our organization, roles are well connected ($\text{Connectedness}_{\text{Coord}}$ value equal to the optimum) while avoiding ambiguities ($\text{Univocity}_{\text{Power}}$ value approaches the optimum). An optimal robustness would require a complete connectivity between all nodes. Due to the privacy and local management policy, this complete connectivity is not possible in a grid context. For example each resource in a domain has its own management policy and doesn't authorize the Global Scheduler to interact directly with it.

Flexibility of an organization is related to the ability to easily adapt. So, in order to enhance the flexibility, a low degree of connectedness for the power and a high degree for the coordination dimensions are needed. Considering the connectivity aspect, while the power relation restrains the flexibility (it imposes constraints on the interactions), the coordination one eases the interactions: we reach a trade-off (see Table 2). Considering the economy aspect, we have an average value which corresponds to a moderate flexibility. To obtain a lower value of the $\text{Economy}_{\text{Coord}}$, for more flexibility, we should increase the redundancy of the coordination which is incompatible with the privacy constraint of the grid context.

Efficiency of an organization can be obtained if it is economic (value equal to 1) in all the dimensions. In our case, the roles are well connected while avoiding redundant ones and we approach the optimal efficiency. The $\text{Economy}_{\text{Control}}$ is equal to the optimum one since we don't take into consideration this dimension.

It is important to notice that the framework proposed by Grossi [10], used for analyzing organizational structures (OS), does not take into consideration the number of roles' instances and the links among them. However, in our context, introducing the roles' cardinalities could be more significant and realistic. For example, a broken resource doesn't imply that the resource role dysfunctions, but means only that this specific instance is out of work and therefore, it could be replaced by one another.

Also, obviously, designing an OS that maximizes simultaneously the three qualities cannot be reached. For example, the more the OS' nodes are well and directly connected, the more the OS may be robust and flexible and the less it is efficient.

Since our organization is devoted to the grid scheduling system, we believe that the qualities obtained (sufficiently robust and efficient) are, in our opinion, the most useful for this context.

As our context is highly dynamic and characterized by a frequent resources performances fluctuations and crashes, we are also interested in the capacity of our organization to be *adaptive* (ability of the system to react against the environment changes). The adaptativity cannot be measured using the previous framework. In the next section, we will measure the quality of our model experimentally by simulations and more precisely its *adaptability*. For this purpose, we will compare the nominal functioning of our system and its functioning with disturbances.

3.2.2 Statistical Evaluation

In this section, we describe our experiments and results.

We have implemented a Grid Scheduling Simulator based on the AGR meta-model using Java and the Madkit [12] platform. Our simulator offers the following services:

- It allows users to enter parameters for the simulations (resources with their numbers, capacities, organization in domains, resource's failure probability, applications' tasks and their coordination process, submission time, etc).
- It simulates the applications' tasks scheduling and their executions step by step,
- It finally outputs the statistical data of the performance metrics on a log file.

Our experimental system is configured with twelve agents, illustrated by the figure 3. The Global Schedulers are represented with a square (GS1, GS2), the Local Schedulers are represented with an ellipse and resources with a rounded rectangle (including the CPU speed). The following grid specification is simulated and organized according to our proposed AGR model described in section 2.2.

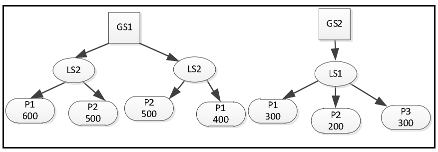

Fig. 3. Case study: agent's organization

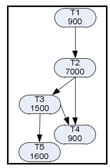

Fig. 4. Case study: application description

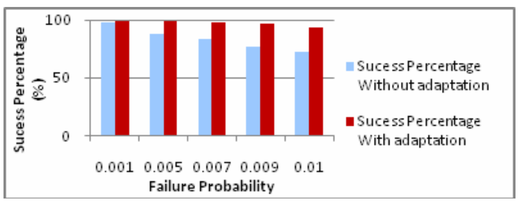

Fig. 5. Execution success percentage

The goal of these simulations is to demonstrate the feasibility (figures 5) and the efficiency (figure 6, 7, 8 and 9) of our model and more precisely of its adaptation.

To demonstrate *the feasibility of our proposition*, we compute the number of successful executions of applications in the case of adaptation comparing to the case with no adaptation. To do that, we consider four same applications (figure 4 shows the application description) and we vary the resources' failure probability. We measure the percent of successful runs. The total number of run is fixed to 100.

The results show the efficiency of our adaptation model since we obtain better percentage of successful executions with adaptation. We can however notice that we do not reach 100% of success since even with adaptation we can meet situations where some tasks couldn't be executed by lack of specific resources.

To demonstrate the *efficiency of the adaptation* of our model, we make two experiments. In the first one, we consider only one application and we vary the resource's failure probability. Figure 6 compares the application execution time with and without disturbances. The results show that the adaptation phase is not costly (at the maximum 27% of the application execution time). Figure 7 shows how the number of communication' messages increases as the failure probability raises. This evolution is due to the fact that adaptation is mainly based on interactions.

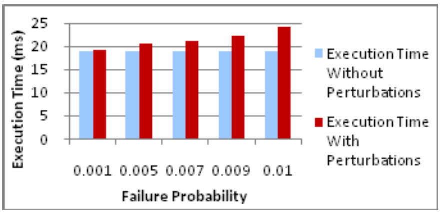

Fig. 6. Comparison of application execution time

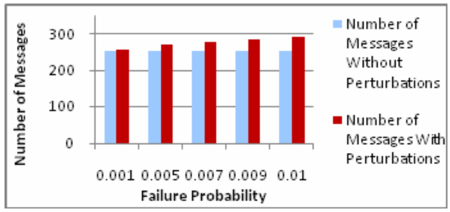

Fig. 7. Comparison of number of messages

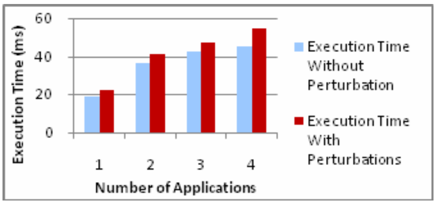

Fig. 8. Comparison of application execution time

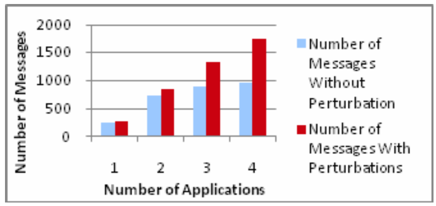

Fig. 9. Comparison of number of Messages

The second experiment varies the number of concurrent applications executed by the system with a fixed resource's failure probability (0.01). Figure 8 shows again that the adaptation time remains low (20% of the execution time) even if the number of applications increases. Figure 9 shows that the number of messages increases with the number of applications in both cases. In case of perturbations, we notice a higher but reasonable number of communications due to the adaptation phase.

4 Conclusion

We have proposed an organizational model to describe and implement a grid scheduling system. This system has been validated: it has been implemented and its design and performances have been discussed. Following the Grossi's framework we have evaluated its conceptual aspect and shown that it is sufficiently robust and efficient. According to our implementation, we have shown the feasibility and efficiency of our approach and more precisely measured the adaptability of our organizational model. In addition, the proposed system presents the following qualitative advantages:

Easy design and implementation of the grid scheduling system. Conventional design of such a system is known to be a difficult task due to the constraints of the environment. The organizational perspective, followed in this paper, constitutes a design support and makes it possible to structure the overall functioning through the attribution of roles and interaction rules to which the agents must conform.

Openness. The use of an organizational perspective eases openness: agents playing predefined roles can enter or leave the grid freely.

Reusable framework. Our work is reusable and could be considered as a first step towards a simulation framework for designing and testing grid organizations. Indeed, designers could use our system to model their own organizations, measure their qualities (robustness, flexibility, efficiency), tune their organizations in order to fit with their requirements, validate these qualities and finally simulate their functioning.

This work opens three main issues for future research. The first issue we are currently investigating is how to add the *control dimension* and what are the effects on the qualities of our organization. The second issue is related to the *structural evaluation* of our organization. We do believe that a refinement of the theoretical evaluation is needed. Indeed, taking into account the roles cardinalities and the relation's occurrences will certainly influence the evaluation of our model. Finally, our simulations were only limited to four concurrent applications. To measure the *scalability* of our approach, we need to consider more applications and possibly move from a simulation framework to a real grid environment.

References

1. Kesselman, C., Foster, I.: The grid: blueprint for a new computing infrastructure. Morgan Kaufmann, San Francisco (2004)
2. Foster, I., Kesselman, C., Jennings, N.: Brain Meets Brawn: Why Grid and Agents Need Each Other. In: 3th International Joint Conference on Autonomous Agents and Multi-Agent Systems, pp. 8–15. IEEE Computer Society, Los Alamitos (2004)
3. Shi, Z., Huang, H., Luo, J., Lin, F., Zhang, H.: Agent-based grid computing. Applied Mathematical Modelling 30, 629–640 (2006)
4. Patel, J., Teacy, W.T., Jennings, N.R., Luck, M., Chalmers, S., Oren, N., Norman, T.J., Preece, A., Gray, P.M., Shercliff, G., et al.: Agent-based virtual organisations for the grid. Multiagent and Grid Systems 1, 237–249 (2005)
5. Muthuchelvi, P., Anandha Mala, G.S., Ramachandran, V.: Agent Based Grid Resource Discovery with Negotiated Alternate Solution and Non-Functional Requirement Preferences. Journal of Computer Science 5, 191–198 (2009)
6. Berman, F., Wolski, R., Casanova, H., Cirne, W., Dail, H., Faerman, M., Figueira, S., Hayes, J., Obertelli, S.G.J.: al.: Adaptive computing on the grid using AppLeS. IEEE Transactions on Parallel and Distributed Systems 14, 369–382 (2003)
7. Ferber, J., Gutknecht, O., Michel, F.: From agents to organizations: An organizational view of multi-agent systems. In: Giorgini, P., Müller, J.P., Odell, J.J. (eds.) AOSE 2003. LNCS, vol. 2935, pp. 214–230. Springer, Heidelberg (2004)
8. Dignum, V.: The Role of Organization in Agent Systems. Multi-agent Systems: Semantics and Dynamics of Organizational Models. IGI (2009)
9. Ferber, J., Gutknecht, O.: A meta-model for the analysis and design of organizations in multi-agent systems. In: The International Conference on Multi Agent Systems, pp. 128–135 (2002)
10. Grossi, D., Dignum, F., Dignum, V., Dastani, M., Royakkers, L.: Structural evaluation of agent organizations. In: 5th International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 1110–1112. ACM, New York (2006)
11. Kaddoum, E., Gleizes, M.P., Georgé, J.P., Picard, G.: Characterizing and evaluating problem solving self-* systems. In: Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, pp. 137–145 (2009)
12. Madkit Website, `http://www.madkit.org/`
13. Thabet, I., Hanachi, C., Ghédira, K.: Towards an adaptive grid scheduling: Architecture and protocols specification. In: Håkansson, A., Nguyen, N.T., Hartung, R.L., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2009. LNCS, vol. 5559, pp. 599–608. Springer, Heidelberg (2009)
14. Smith, R.G.: The contract net protocol. IEEE Transactions on High-level communication and control in a distributed problem solver 100, 1104–1113 (2006)