Using Trusted Communities to Improve the Speedup of Agents in a Desktop Grid System

Lukas Klejnowski, Sebastian Niemann,
Yvonne Bernard, and Christian Müller-Schloer

Abstract. In open, technical, multi-agent based systems, self-interested agents can show behaviours that degrade the performance of the system. This can be countered by providing cooperation incentives. In this paper, we present a formalisation of delegation incentives for an open, agent-based, Desktop Grid system, based on decision trees. We then discuss reputation-based delegation strategies, as well as replication-based delegations strategies and focus on the incentives these strategies provide for agents to cooperate. We further show why we see room for improvement, and how this can be achieved with organisation-based delegation. We propose a delegation strategy based on *Trusted Communities* and present evaluation results for the comparison of these strategies with respect to the achieved average speedup in the system.

1 Introduction

In technical, multi-agent-based scenarios, the utility functions of the agents can be used to model the performance of the system, as well as the performance of the single involved participants. However, when these systems are open, we need to consider self-interested agents that do not care for the performance of other agents or the system as such. These agents have to be treated as blackboxes regarding their behaviour, as we cannot demand cooperative types of behaviour without violating agent autonomy. Instead, incentive mechanisms are used to influence the agent behaviour in a way that increases their willingness to cooperate.

In this paper, we discuss incentive mechanisms for such an open, multi-agent-based, technical system and argue how incentive mechanisms based on (a) trust and reputation, and (b) trust-based MAS-organisations, can be used to improve the performance of this system.

Lukas Klejnowski · Sebastian Niemann · Yvonne Bernard · Christian Müller-Schloer
Leibniz Universität Hannover, ISE-SRA, Appelstr. 4, Hannover, Germany
e-mail: {klejnowski,niemann,bernard,cms}@sra.uni-hannover.de

F. Zavoral et al. (eds.), *Intelligent Distributed Computing VII,*
Studies in Computational Intelligence 511,
DOI: 10.1007/978-3-319-01571-2_23,

2 System Model

We study an open, distributed and volunteer-based Desktop Grid System (DGS) to which we refer as *Trusted Computing Grid (TCG)*, in the tradition of systems like XtremWeb [14]. The system is designed without central control and the applications regarded produce bag-of-task jobs, i.e. jobs being composed of independently processable work units (WUs). Each user can submit jobs to the system and is expected to volunteer its machine as worker for other users' work units.

We use agents that are in charge of the grid client on the machines and make decisions on behalf of their users. The utility of the agents is based on their goal, to schedule single work units on available worker agents, such that they minimise the time it takes to receive valid results. This is formalised using metrics like *speedup*, *processing time* and *waste* (cf. e.g. [11] and [4]). Due to the open nature of the system, we have to deal with agents that show various types of behaviour (from altruistic to untrustworthy) in order to achieve their self-interested goal of scheduling their own jobs as efficiently as possible. According to the taxonomy of [10] and taking the resource perspective, we therefore classify the potentially participating agents of this Desktop Grid System as: egoistic, volatile, distributed over the internet, dynamic, faulty and heterogeneous. We therefore apply a Trust- and reputation system and let the agents build up trust relationships based on the outcome of their interactions as submitters and workers.

2.1 Submitter Decision Tree

In the following we present the agents' decision tree as depicted in Fig. 1, when taking the submitter perspective and searching for suited worker agents. An agent x from the agent society A that has an unprocessed work unit τ, first builds up the set $Y(x)$, containing each agent y that qualifies as worker based on its estimated performance P^e_y. P^e_y is dependent on the machine performance, the host and resource availability, and the work load of agent y. In sum, it represents an estimate of y's *competence* as a delegation partner for x. The set $Y(x)$ is formally composed as follows:

$$Y(x) := \left\{ \widetilde{y} \in A : P^e_{\widetilde{y}} > P^e_x \right\} \tag{1}$$

The submitter then makes a decision $D(x)$ which delegation strategy is suited best for delegating τ to a worker agent $y \in Y(x)$. This is based on an assessment of y's *willingness* to cooperate, being the probability that y is willing to invest an effort e. In a DGS, the positive outcome o^+ of the delegation is a state, where a valid result for τ exists, which can only be reached if y processes τ completely (effort e^+). Each other effort level produces a negative outcome o^-, meaning no or no valid result for τ. Here we discern between straight rejection to cooperate (e^0) and the attempt to produce o^+ through processing τ to a certain degree d, denoted by e^-_d. Additionally, y can be a malicious agent, choosing e^-_d to harm x. Whenever the outcome o^- was reached, x enters a new round r and either delegates to the next best worker in $Y(x)$ or processes τ itself when no agent is left in $Y(x)$. This is done until the outcome

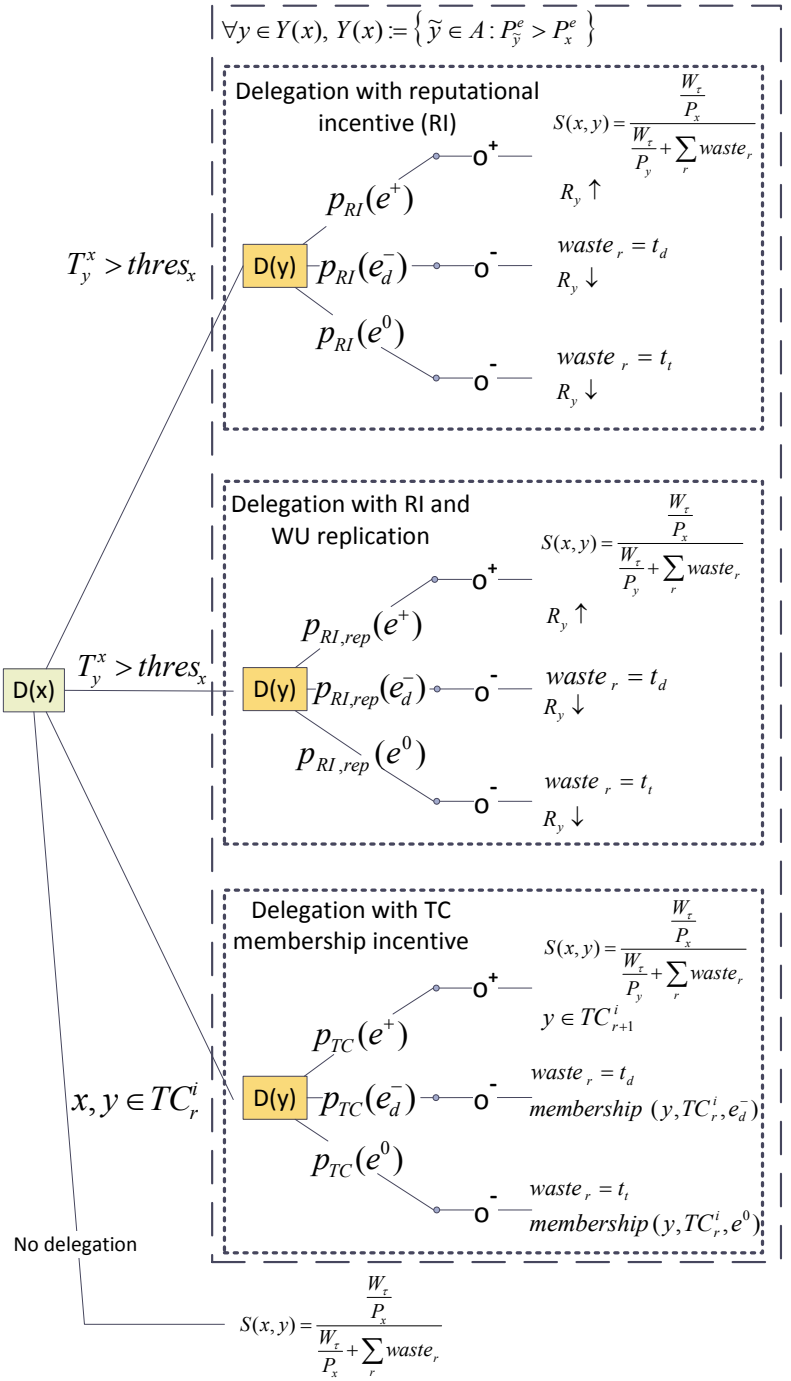

Fig. 1 Decision tree for submitter agents

is o^+, with the assumption that x always produces o^+ when processing an own τ, hence always terminating.

In the following, we argue how x can choose a delegation strategy and what implications that choice has on the utilities and costs for both x and y. In the *TCG*, the utility of an agent is defined by the average *speedup* it is able to achieve for each task τ it submits. The speedup is defined as follows:

$$S(x,y) = \frac{processingtime_{x,\tau}}{processingtime_{y,\tau} + waste} = \frac{\frac{W_\tau}{P_x}}{\frac{W_\tau}{P_y} + \sum_r waste_r} \tag{2}$$

In eq. 2 we incorporate the size W_τ of a work unit and the actual worker performance P_y, and contrast this with the a posteriori measure of the submitter performance P_x and the summed waste in all rounds r with the outcome o^-. The $waste_r$ is the time that τ was in processing in round r without an obtainable result[1]. In case of e_d^- the amount of waste depends on the time t_d spent with the effort e_d^-, while the waste generated by the processing rejection e^0 is the time t_t needed for the transition to the next round r+1 and try to delegate τ to the next worker. In general, we observe $t_d \gg t_t$. It is obvious, that agents need to interact with cooperating workers as much as possible to increase their speedup.

The option to *not delegate* τ and process it by itself is always possible for x, but is only used as last resort when no competent or willing workers were found. This is because the speedup is dependent on x's own performance P_x and can therefore never be greater than 1 for this strategy.

As long as there are willing workers available, the first delegation strategy with *reputation incentive* is preferred over self processing: We use a trust- and reputation system to rate the workers according to their effort[2]. Positive outcomes result in y's reputation gain (denoted by $R_y \uparrow$) and negative outcomes in the loss of reputation $R_y \downarrow$, with the amount of loss being dependent on the costs for x. We then let each agent x define a trust threshold $thres_x$ such that the subjective trustworthiness T_y^x of y has to be greater than that threshold in order to consider y as suited worker. But how does a high effort pay off for the workers? The answer is through reciprocity: A worker y will make the decision $D(y)$ to reject a cooperation with x, if $T_x^y \leq thres_y$, thus if itself would not consider x as a suited worker for its own work units[3]. In this way reputation acts as incentive to cooperate and produces reputation loops between worker and submitter performance.

The second delegation strategy with *WU replication* (cf. Fig. 1) also applies the reputation incentive, however, here submitters try to decrease the probability for a low speedup to the disadvantage of the system: Instead of waiting until the result

[1] This is true for applications that do not allow checkpointing.

[2] In order to do this, we need to discern the outcomes o^- and o^+. This is only possible if we have applications that produce work units whose results can be validated with far less effort than generated.

[3] Agent y will also reject if its work load is already too high in order to maintain a condition in which it could process own work units with low costs if necessary.

for τ produced by a worker y can be validated, x generates replicas τ_i of τ and tries to delegate these to other workers. As soon as a valid result is returned, it can cancel the processing of the remaining replicas. This is an appropriate strategy from the cost perspective of a submitter, as replication generates hardly any additional overhead. However the work load in the system is raised by the replication factor until o^+ is reached: This not only reduces the probabilities $p_{RI,rep}(e^+)$ and $p_{RI}(e^+)$, as $D(y)$ depends on y's work load, for other submitting agents, but also for x itself, as work units τ come in bursts (jobs). In the long term the speedup of x can therefore even decrease. On the worker side, wasteful processing of replicas blocks the worker. This reduces the opportunities to work for agents that could reciprocate and hence counters the effects of the replication incentive. When applying this strategy, submitting agents should therefore take the work load of suited workers into account when making the decision $D(x)$.

2.2 Discussion

Delegation with reputation incentives introduces a problem addressed in [7]: With agents accumulating trustworthiness through cooperation, a situation can develop where there is too much trust (*over-confidence*) to quickly react to changing behaviour. This becomes clear when regarding $thres_x$: A highly trustworthy agent can have a strong negative impact on the speedup of other agents, if it starts to defect, because it will only be ruled out as worker if successive reputation losses lower its trustworthiness below the threshold. Additionally, too much trust in a worker y means that a submitter x subsequently invests more validation and monitoring costs to evaluate the worker than effectively necessary.

We therefore argue that we need an additional delegation strategy that counters these issues and hence allows for faster reaction to changing behaviours of trustworthy agents and lower costs for submitters. Additionally, the incentive provided by this new delegation strategy has to be stronger than the other incentives, in order to enable worker agents to participate in these interactions. In the following, we propose such a strategy, with delegation based on the membership in an organisation called *Trusted Community*, as introduced in [19].

2.3 Delegation within Trusted Communities

Trusted Communities (TCs) are formed and joined by self-interested agents with strong mutual trust relations and the purpose to increase their personal utility. TCs are maintained by management actions delegated to a designated member called *Trusted Community Manager (TCM)*, having the goal to preserve and optimise the composition and stability of this organisation. This organisation provides performance benefits for their members by improving interaction efficiency, information sharing and cooperation between the agents.

Agents become members of a TC depending on the strategy the TCM has with respect to the composition. In general, the realisation of this strategy is based on the

observed trustworthiness of agents in the society. When becoming a member of a TC TC_r^i, an agent y makes a contract with the TCM and all other members. This contract is based on the notion of *kinship* and states that, if chosen as delegate by an other member x in the round r, the agent y commits to provide the effort e^+. In case of a contract violation due to effort e^0 or e_d^-, the TCM applies a *member control strategy* on the TC_r^i, such that it decides on the membership of y, formalised by:

$$TC_{r+1}^i = membership(y, TC_r^i, e^0) \in \left\{TC_r^i, TC_r^i \setminus \{n\}\right\}^4 \tag{3}$$

This contract provides strong incentives to invest the effort e^+, as agents can rely on other members also providing them with the same effort. On the other hand, the validation of a contract violation through e^0 is cheap in terms of costs and fast as no processing time is involved. The validation of a violation via e_d^- is obviously more expensive, but can be fairly distributed among the members of a TC. In sum, we thus expect $p_{TC}(e^+) > p_{RI}(e^+)$. This means that the usage of a TC-based delegation strategy is more suited to increase the speedup of agents, than the usage of a reputation-based delegation strategy, as will be shown in section 3. Additionally, this incentive mechanism avoids the problems of too much trust and over-confidence, as we do not need to rate workers with trust- and reputation values, but can react to uncooperative behaviour immediately (with membership loss).

Besides controlling members in the worker role, the *TCM* can control members in the submitter role. This is realised by a monitoring scheme that is transparent to the submitters and does induce only low costs on the members: Workers in the TC report information on the work units accepted for processing to the *TCM*, which then is able to detect whether submitters have applied WU replication. Again this can be sanctioned via the *member control strategy* and works as incentive to cooperate. This monitoring is of course not applicable in general in the whole system, but needs a scaled environment like the TC.

The applicability of the *Trusted Community* delegation strategy is dependent on the trust-relationships within the agent society: Only where mutual trust builds up through the reputation incentive delegation strategy, TCs can be formed. Besides, these relationships often develop between clusters of agents that can partition the society into groups of cooperating agents. We consider this by allowing the formation of several Trusted Communities. In the following, our evaluation of the benefit of the delegation strategies is presented.

3 Evaluation

We have conducted experimental evaluations in simulations of the Trusted Computing Grid, to show that the formation and operation of Trusted Communities as delegation control organisation can improve the speedup of the member agents and hence the average speedup of cooperative agents in the system. As experimental setup we have used an agent society with 100 agents with heterogeneous machine

[4] Analogous for effort e_d^-.

performance and behaviour with respect to the preferred effort level (adaptive to system perception, free-riding or selfish). Each agent produced in average 21 grid jobs composed of an average of 16 WUs. Thus the minimal number of *D(s)*-decision evaluations was 33600 during the experiment runtime. The experiments were conducted with the option to have either no TC formation, allow for a single TC to form, or no restriction to the number of formed TCs. In the case of no TC formation, agents have used the delegation strategy with the reputation incentive. We then have conducted 20 runs per option and measured the variance in the achieved speedup as depicted in Fig. 2.

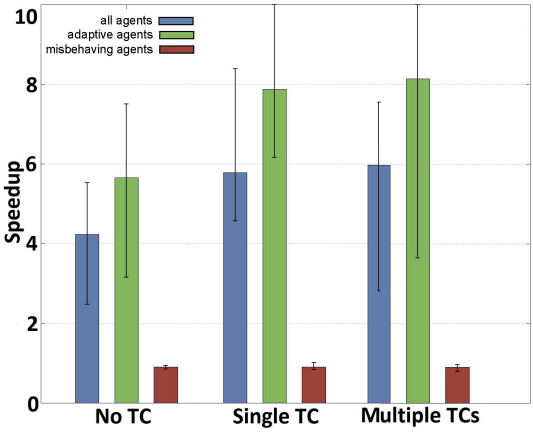

Fig. 2 Speedup gain through the application of Trusted Communities

The results show that, due to the TC membership incentive based on the kinship contract, TCs are able to increase the speedup of member agents. Additionally, the formation of several TCs is suited to further improve the results which shows the scalability of the approach. The variation of the single run results is relatively high, indicating that the composition and performance of the TCs very much depends on the system state in the *TCG*. However, we observe also a high variation for the system without TC formation, such that we attribute this to the varying agent behaviour dynamics. What the incentives have in common is, that notoriously uncooperative agents are isolated, such that they are forced to apply the *no delegation* strategy, resulting in a low speedup.

4 Related Work

We base our methodology on the generic decision tree for trustor agents presented in [6] and extend this work by applying it to the Desktop Grid domain, as well as propose an additional delegation strategy based on our work on the multi-agent

organisation *Trusted Community* (cf. [19]). Similar approaches to MAS organisations can be found in [16], [5] and [20]. In these approaches, we find that some requirements for the open Desktop Grid System we evaluate here are not met. In particular, *congregations* by [5] do not incorporate any notion of trust. In open distributed computing systems, trust and reliability play an essential role in the partitioning of participants into reliable and unreliable hosts (cf. e.g. [13], [3], [21]). *Clans*, as presented in [16], are best described as *congregations with trust* (cf. [18]). As such, clans provide the previously mentioned support for trust-based interaction modelling. However, clans are a purely decentralised approach without hierarchy. We assess this to be detrimental in systems environments where agent behaviour is highly dynamic, as we assert single agents the potential to damage the stability of an MAS-organisation if no coordinated self-management is performed. We therefore incorporated hierarchy in the TC design. Additionally, a strict reliance on a trust management system can also be detrimental if over-confidence builds up. In this we follow the argumentation in [7] regarding negative consequences of over-confidence. The way we implement TC membership as favourable compared to non-association is based on the ideas of incentive compatible design (cf. e.g. [22]). We especially follow the argumentation that agents are rational with respect to maximising their utility function (cf. e.g. [8]) and refer to a concrete utility definition (*speedup*, cf. e.g. [11],[23]) for an open Desktop Grid. The incentives for Desktop Grid agents provided by TC membership are a reciprocity-based approach, comparable to approaches discussed in e.g. [15], however transferred to agents representing the users. Again, the conclusion by the authors that these types of approaches are vulnerable to collusive behaviour, confirms our approach of hierarchical management of a TC as introduced in [19] and scheduled for future work. As for our application scenario, the main classification according to the system perspective in the taxonomy presented in [10], is that of *volunteer-based, distributed, P2P-based, internet-based Desktop Grid System*. Additionally, taking the resource perspective, we can conclude that participants are *egoistic, volatile, distributed over the internet, dynamic, faulty and heterogeneous*, comparable to e.g. the approach presented in [9]. This constitutes a competitive system with great uncertainty regarding expected performance and can be addressed by the application of trust as incentive-criterion, as for example in [12], [23], [17] and [13].

Additional coverage of MAS organisations in similar scenarios has also been considered in [24], [1] and [25].

5 Conclusion and Outlook

We presented an approach to formalise the delegation decision of agents in an open Desktop Grid System and described how a delegation strategy with an reputation incentive can improve the average speedup of these agents. We then discussed the drawbacks of this approach and proposed a new approach based on the application of the MAS-organisation *Trusted Community*. We showed how this delegation strategy can counter the drawbacks of a pure reputation strategy and presented the

evaluation results. These show that the application of TCs leads to a higher average speedup in the system. Future work will focus on exploiting this incentive mechanism further by using trustworthiness prediction, as in [2], to allow for earlier TC formation and thus less over-confidence. Additionally, we will compare the results with the performance of a delegation within *clans*.

Acknowledgements. This research is funded by the research unit "OC-Trust" (FOR 1085) of the German research foundation (DFG).

References

1. Abdallah, S., Zhang, H., Lesser, V.: The role of an agent organization in a grid computing environment. In: Proceedings of the 14th Int Conference on Automated Planning and Scheduling, Workshop on Planning and Scheduling for Web and Grid Services (2004)
2. Anders, G., Siefert, F., Steghöfer, J.-P., Reif, W.: Trust-Based Scenarios - Predicting Future Agent Behavior in Open Self-Organizing Systems. In: Proceedings of the 7th Int. Workshop on Self-Organizing Systems, IWSOS 2013 (2013)
3. Andrade, N., Brasileiro, F., Cirne, W., Mowbray, M.: Discouraging Free Riding in a Peer-to-Peer CPU-Sharing Grid. In: Proceedings of the 13th IEEE Int. Symposium on High Performance Distributed Computing, pp. 129–137. IEEE Computer Society, Washington, DC (2004)
4. Anglano, C., Brevik, J., Canonico, M., Nurmi, D., Wolski, R.: Fault-aware scheduling for Bag-of-Tasks applications on Desktop Grids. In: 2006 7th IEEE/ACM Int. Conference on Grid Computing, pp. 56–63. IEEE (2006)
5. Brooks, C., Durfee, E.: Congregation formation in multiagent systems. Autonomous Agents and Multi-Agent Systems 7(1) (2003)
6. Burnett, C., Norman, T.J., Sycara, K.: Trust decision-making in multi-agent systems. In: Proceedings of the 22nd Int. Joint Conference on Artificial Intelligence, vol. 1, pp. 115–120 (2011)
7. Castelfranchi, C., Falcone, R.: Trust Theory - A socio-Coginitive and Computational Model. John Wiley & Sons Ltd. (2010)
8. Centeno, R., Billhardt, H.: Using incentive mechanisms for an adaptive regulation of open multi-agent systems. In: Proceedings of the 22nd Int. Joint Conference on Artificial Intelligence, Barcelona, Spain, vol. 1, pp. 139–145 (2011)
9. Chakravarti, A., Baumgartner, G., Lauria, M.: The organic grid: self-organizing computation on a peer-to-peer network. In: Proceedings of the Int. Conference on Autonomic Computing, pp. 96–103. IEEE (2004)
10. Choi, S., Buyya, R., Kim, H., Byun, E.: A Taxonomy of Desktop Grids and its Mapping to State of the Art Systems. Technical report, Grid Computing and Distributed Systems Laboratory, The University of Melbourne (2008)
11. Cremonesi, P., Turrin, R.: Performance models for desktop grids. In: Proceedings of the 10th Int. Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS). Citeseer (2007)
12. Domingues, P., Sousa, B., Moura Silva, L.: Sabotage-tolerance and trustmanagement in desktop grid computing. Fut. Gener. Comput. Syst. 23(7) (2007)
13. Dyson, J., Griffiths, N., Lim, H., Jarvis, S., Nudd, G.: Trusting agents for grid computing. In: 2004 IEEE Int. Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583) (2004)

14. Fedak, G., Germain, C., Neri, V., Cappello, F.: XtremWeb: A generic global computing system. In: IEEE/ACM Proceedings of the 1st Int. Symposium on Cluster Computing and the Grid. IEEE Computer Society (2001)
15. Feldman, M., Chuang, J.: Overcoming free-riding behavior in peer-to-peer systems. ACM SIGecom Exchanges 5(4), 41–50 (2005)
16. Griffiths, N.: Cooperative clans. Kybernetes 34(9/10) (2005)
17. Griffiths, N.: Task delegation using experience-based multi-dimensional trust. In: Proceedings of the 4th Int. Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2005, p. 489. ACM Press, New York (2005)
18. Horling, B., Lesser, V.: A Survey of Multi-Agent Organizational Paradigms. The Knowledge Engineering Review 19(4), 281–316 (2005)
19. Klejnowski, L., Bernard, Y., Anders, G., Müller-Schloer, C., Reif, W.: Trusted Community - A Trust-Based Multi-Agent Organisation for Open Systems. In: Proceedings of the 5th Int. Conference on Agents and Artifcial Intelligence (ICAART), Barcelona, Spain (2013)
20. Mathieu, P., Routier, J.-C., Secq, Y.: Principles for dynamic multi-agent organizations. In: Kuwabara, K., Lee, J. (eds.) PRIMA 2002. LNCS (LNAI), vol. 2413, pp. 109–122. Springer, Heidelberg (2002)
21. Messina, F., Pappalardo, G., Rosaci, D., Santoro, C., Sarné, G.M.L.: A Trust-Based Approach for a Competitive Cloud/Grid Computing Scenario. In: Fortino, G., Badica, C., Malgeri, M., Unland, R. (eds.) Intelligent Distributed Computing VI. SCI, vol. 446, pp. 129–138. Springer, Heidelberg (2012)
22. Ramchurn, S.D., Huynh, D., Jennings, N.R.: Trust in multi-agent systems. The Knowledge Engineering Review 19(01), 1–25 (2004)
23. Shudo, K., Tanaka, Y., Sekiguchi, S.: P3: P2p-based middleware enabling transfer and aggregation of computational resources. In: Proceedings of the IEEE Int. Symposium on Cluster Computing and the Grid CCGrid 2005, vol. 1 (2005)
24. Thabet, I., Bouslimi, I., Hanachi, C., Ghédira, K.: A multi-agent organizational model for grid scheduling. In: O'Shea, J., Nguyen, N.T., Crockett, K., Howlett, R.J., Jain, L.C. (eds.) KES-AMSTA 2011. LNCS, vol. 6682, pp. 148–158. Springer, Heidelberg (2011)
25. Wang, Y., Vassileva, J.: Trust-based community formation in peer-to-peer file sharing networks. In: Proceedings of the 2004 IEEE/WIC/ACM Int. Conference on Web Intelligence, WI 2004. IEEE Computer Society, Washington, DC (2004)