![InterNeg logo]

Decision Station for Service Level Agreement Negotiations in Computing Grids

Tobias Heitfeld[1], Dirk Neumann[1], Rustam Vahidov[2]

[1] Institute of Information Systems and Management (IISM), Universität Karlsruhe (TH)
[2] John Molson School of Business, Concordia University

Abstract

Recently, the grid computing paradigm has been gaining considerable attention among the industry practitioners and the researchers in the field. One of the central issues in grid computing is effective allocation of computing tasks among the nodes in the grid. In this paper a negotiation-based approach is considered for deriving service level agreements (SLAs) among the participants. The computing resource providers could thus sell their resources through multiple negotiation processes taking place in parallel. The paper proposes applying "situated decision support" (or "decision station") approach to manage SLA negotiations by a single provider. Situated decision support systems effectively combine human judgment with autonomous decision making and action by agents. The intuition of using this approach for SLA negotiations lies in the monitoring and controlling of the fleet of local agents negotiating single services from multiple service providers by the use of a "manager" agent and human decision maker. The paper describes the approach and presents the results of simulation experiments.

1. Introduction

Grid computing refers to a computing model that distributes processing across an administratively and locally dispersed computing infrastructure. The ability to connect numerous heterogeneous computing devices, allows the creation of virtual computer architectures from otherwise idle resources. According to market surveys is the average productivity of IT infrastructures extremely low: Mainframes are up to 40 % of the time idle; UNIX servers even more than 90 % of the time not used; individual PCs reach a utilization rate of at most 5 % (Carr 2005). This daunting ineffectiveness of IT infrastructures and the associated costs entail that enterprises seek to use Grid technologies to outsource IT services. Rather than investing and maintaining proprietary infrastructures enterprises tap to the Grid and consume IT services on-demand. Service providers like SUN and Amazon accommodate this need by offering CPU hours and storage over web-services.

IT services outsourcing, however, comes with the associated risks (Carr 2005). Service providers address this risk by referring to formal contracts, so-called Service Level Agreements (SLAs). SLAs define the common understanding about services (e.g., priorities, responsibilities, guarantees and penalties once SLAs are violated) between service providers and consumers. Essentially, SLAs are the crucial instrument for service consumers to formulate guarantees on the service quality delivered by the service provider. Nonetheless, SLAs are also an important tool for service providers to advertise free service capacities as well as to manage their internal resources.

In light of the above considerations SLA management is currently an important area of research (Hasselmeyer et al. 2006; Deora and Rana 2007). One major success has been the specification of the de-facto standard *WS-Agreement* as a description language for SLAs by the Open Grid Forum (OGF). WS-Agreement allows the formulation of well defined and comprehensive contracts, which minimizes the risks of disputes resulting from unclear formulations among service provider and consumer. Clearly, a description language is merely prerequisite for well defined SLAs. What is equally important is a procedure with which SLAs are created. This becomes even more important, if SLAs are dynamically adapted dependent on resource availability and demand. If the SLAs are well-negotiated, the likelihood that balanced SLAs are signed is much higher. In this spirit, the Grid Resource Allocation Agreement Protocol Working Group (GRAAP-WG) (http://forge.gridforum.org/projects/graap-wg) of OGF set on their agenda for future work to define so-called negotiation profiles or protocols for SLA negotiation as part of the WS-Agreement specification (Rana 2006).

Negotiating the terms of SLAs can be quite complex as service consumers need to negotiate with multiple service providers to reserve different resources. Likewise, service providers need to effectively negotiate SLAs with multiple service buyers in light of market conditions, availability of resources, company policies, and other factors. Due to this complexity the use of software agents have been proposed for managing SLA negotiations (Czajkowski et al. 2002; Ouelhadj et al. 2005; Eymann et al. 2006).

In this work our interest is in one-to many SLA negotiations involving one service provider and multiple potential customers. In such settings the complexity involved in tracking and management of multiple on-going negotiations can be alleviated by means of intelligent support (e.g. software agents). However, in our opinion, a balanced approach combining

autonomous action with overall human control and decision-making is a more adequate approach, as it allows for timely intervention and improves the predictability of business outcomes. In this respect the recently introduced model of "situated decision support" seems to be a promising framework to apply to SLA negotiations (Vahidov 2002; Vahidov and Kersten 2004).

Situated decision support advocates a "connected" and active mode of decision support and implementation. The major components of situated decision support system (also called "Decision Station") include sensors, effectors, manager, and active user interface. In this setup, decision support expands to include problem sensing and action generation and monitoring of the implementation of decisions in addition to the conventional intelligence/design/choice phases. It also allows flexibility for effectively combining autonomous action by the system with judgmental input from the human decision makers.

Since the fields of decision and negotiation support are closely related, in this work we show the potential applicability of the situated decision support model to managing multiple concurrent SLA negotiations, where a managing entity monitors and controls the local negotiation agents. In this paper we elaborate on the application of the Decision Station (DS) concept to managing multiple SLA negotiations by the service providers, present the system prototype, and report the results of simulation experiments.

2. Service Level Agreements

A service level agreement is defined a contract between a service provider and consumer that specifies the rights and obligations of the provider and the penalties that will be applied if those obligations are not satisfied. Since the late 80s SLAs have been used by telecom operators as contracts with their corporate customers. With the recent trend of outsourcing and application service providing, IT departments have adopted the idea of using service level agreements with their internal or external customers.

SLAs for Grid computing are not very different than those for other services (e.g. computation, or storage services). But while the elementary issues of an SLA can be the same, negotiating SLAs for Grid services is aggravated by the fact that Grid services are typically composed of several basic services. This implies that it is essential needs to manage concurrent SLA negotiations with multiple service providers (Sahai et al. 2003). A snapshot of typical attributes of SLAs for Grid is shown in Table 1.

Table 1 Example of SLA attributes

Attributes	Attribute Levels
Service Availability	99,999 % 99,99 %

	99,9 %
Price	< 1,000 €/month
	< 5,000 €/month
	< 10,000 €/month
	> 10,000 €/month
CPU utilization	1,000 CPU hours
	5,000 CPU hours
	10,000 CPU hours
Storage (RAM)	1 GB
	2 GB
	4 GB
	> 4 GB
Bandwidth	1 Mb
	10 Mb
	100 Mb
	1 Tb
Deployment	Webservice
	Source code
	others

It is commonly accepted that the success of SLA negotiations relates to the specification and implementation of a protocol that creates legally-binding agreements. This protocol needs to describe a set of domain-independent messages, together with an abstract schema, such that it can be exchanged by all negotiators (Kuo, Parkin et al. 2006). Several bargaining protocols that meet those requirements have been proposed by (Czajkowski, Foster et al. 2002; Gimpel et al. 2003; Eymann, Neumann et al. 2006; Parkin et al. 2006; Siddiqui et al. 2006).

The bargaining protocols share a very similar structure but differ in some details. For example the protocol proposed by (Siddiqui, Villazon et al. 2006) involves multiple rounds among service consumers (e.g., co-allocators) and providers (e.g., schedulers) until they reach an agreement. The providers post their offers on request to the consumers, who can either select among the available offers or enter re-negotiation by relaxing some constraints. Another noteworthy effort has been proposed by (Ludwig et al. 2005) who combine the negotiation protocol with a template-based contract generation process. At the end of the negotiation protocol a fully-fledged SLA is generated as WS-Agreement instance.

3. Software Agents for Negotiations

SLA negotiations can involve a number of participants both on the consumer as well as the provider side trying to reach an agreement in multiple bi-lateral interactions. A given provider, for example, could have several on-going negotiations at the same time and has to set the objectives, constraints and strategies in those negotiation instances in such a way that maximizes the achievement of business objectives, while avoiding over-commitment, i.e. promising more than the provider can deliver. Clearly, with the growing complexity of the composite services and increasing customer base, automated means of supporting multiple negotiations could help the providers to better handle SLA negotiation processes. In this respect, employing intelligent agents seems to be an adequate approach. Below we briefly describe some agent-based approaches to automated negotiations, in particular for the case of multi-bilateral settings.

Most of the related work in the area of agent-based negotiation has been devoted to the design of strategies for agents. The fundamental work has been undertaken without considerations to SLA negotiations. Nonetheless, as those approaches are domain-independent, they are highly relevant for SLA negotiations.

Faratin et al. have proposed a "smart" strategy for autonomous negotiating agents (Faratin et al. 2002). Agents following this strategy would try to make tradeoffs in a manner that the newly generated offer is similar to the opponent's last offer, before trying a concession. Another work in this direction seeks to map business policies and contexts to negotiation goals, strategies, plans, and decision-action rules (Li et al. 2006).

While fully automated negotiations may not always be a feasible choice, agents could also act as intelligent assistants, helping the users by providing advice, critiquing user's own candidate offers as well as the offers by an opponent, and generating candidate offers for a user to consider (Kersten and Lo 2003; Chen et al. 2005).

In multi-bilateral negotiations a negotiator may be having several concurrent negotiation processes taking place at the same time and involving multiple opponents. A fuzzy set-theoretic approach to analysis of alternatives in multi-bilateral negotiations has been proposed in (Van de Walle et al. 2001). The authors have considered scenarios involving RFQ sent by one seller to multiple buyers (agents) with the purpose of deciding which potential buyers to negotiate with. They used fuzzy-relational approach to obtain a partial rank-order of the prospective buyers.

A setup where multiple agents negotiate autonomously and one agent is designated as a coordinator has been proposed in (Nguyen and Jennings 2004) and (Rahwan et al. 2002). In (Nguyen and Jennings 2004) a buyer agent runs multiple concurrent negotiation threads that

interact with several sellers. The coordinator agent provides each thread with reservation values and negotiation strategy. The threads report back to the coordinator, which then advises them on the possible changes to reservation values or strategies. In (Rahwan, Kowalczyk et al. 2002) a similar approach has been used including coordinating agents and multiple "sub-buyer" agents. In (Dang and Huhns 2006) a protocol for handling many-to-many concurrent negotiations for internet services has been proposed.

In most of the above models of one-to-many negotiations it is assumed that the party on the one side can only have one agreement as an outcome of negotiations with multiple opponents (i.e. XOR case). In this work we interested in managing multiple SLA negotiations where each negotiation may fail or succeed regardless of others (OR case). In such settings the system could learn from the agreements made recently and take into account other relevant information (e.g. market situation) to direct concurrent negotiations.

4. Situated DSS for SLA Negotiation

Adoption of negotiations as a primary vehicle for conducting daily regular economic exchanges may result in a considerable effort on human decision makers. Employing automated software components to conduct or assist human negotiators in conducting regular SLA negotiations may be a promising solution to achieve significant cost and time savings. However, with automated negotiations there is a potential danger that the overall process and the important business outcomes may become somewhat unpredictable. Moreover, often the course of negotiations depends on other factors, lying outside the domain of the expertise or sensory capabilities of the agents. For example, customer behavior may be heavily affected by the latest important economic, technology-related and other types of events. This calls for a type of solution allowing certain degree of automation, but allowing the control by the user of the overall process.

In light of the above considerations we propose a type of solution that would effectively combine human judgment capabilities with autonomous actions by agents. The key motivation here is to relieve human users from the necessity of being involved in each and every SLA negotiation process. Rather, the system should allow the human decision maker to effectively and efficiently manage the fleet of negotiating agents to meet and maintain higher-level business targets.

The field of decision support systems (DSS) is very much related to the area of negotiations and negotiation support systems (Jarke et al. 1987). In the recent DSS literature there has been important research streams directed towards building "active" and agent-based systems that would transform the original "toolbox" model of DSS into an active participant in the decision-making process, which could perform some decision-related tasks in an autonomous fashion (Rao et al. 1994; Angehrn and Dutta 1998; Hess et al. 2000; Vahidov and Fazlollahi 2004).

One such model introduced recently is known as *"situated decision support system"*, or *"decision station"* (Vahidov 2002; Vahidov and Kersten 2004). Situated DSS looks to combine the benefits of agent technologies and those of decision support systems to facilitate active problem sensing, decision implementation and monitoring in addition to pure decision support. In essence, situated DSS expands the traditional model from purely "problem-solving/decision making" frame to situation assessment and action generation.

Situated DSS is made up from different active (agent) components in addition to the

traditional "toolbox" of data, models, and knowledge. The components include: sensors (for information search and retrieval), effectors (for affecting current state of affairs), manager (for deciding how to handle a particular situation), and active user interfaces (for intelligent interaction with the user).

The composition and interactions of different components of the situated DSS model provides an appealing blueprint for facilitating one-to-many SLA negotiations with limited autonomous action and overall control of the process by a human decision maker. Various negotiation-related tasks can be mapped to specific agents in the model. For example, tasks including monitoring service levels, sensing potential problems, generating alerts, and predicting market trends for SLA could be delegated to the "sensor" agents. Controlling the fleet of negotiating agents by monitoring their progress and instructing them on the adjustments to negotiation strategies, reservation levels and preference structures can be delegated to some extent to the "manager" agent. Human decision maker could set the limits of authority of this agent and intervene if necessary. Each negotiation instance could be delegated to a particular "effector" agent that conducts a given negotiation and reports on the progress.

The model for supporting multiple SLA negotiations is shown in figure 1.The situated DSS model is essentially hierarchical involving three layers. At the bottom layer, which could be called "operational" the agents perform negotiations. They are given the preferences, aspiration and reservation layers and strategies to follow and try to negotiate effectively with a given opponent. The basic cycle of generating an offer by these agents could be described as: retrieve preference structure (possibly updated by the "manager" agent); compare past offer and counter-offer; generate new offer according to adopted strategy. Automated negotiations have been studied extensively in the recent past and thus we do not focus on the detailed design of negotiating agents. One possibility is to employ "smart" strategy by the negotiating agents proposed in (Faratin, Sierra et al. 2002).

The second layer contains the manager agent, which makes use of the traditional DSS components (data, models and knowledge) to manage the negotiating agents. The manager monitors key indicators of the negotiation processes, such as number of agreements reached, proportion of failed negotiations, resource consumption and others and decides whether to intervene or not. For example, if memory resource becomes scarce, then in the next round of SLA negotiations the manager would instruct the agents to stress more the importance of memory resource in the utility calculation. One way to encode the knowledge of the manager could be through "If-Then" (possibly fuzzy) rules. The following simple example from a manager policy is the following rule which expresses that if memory units sold is much larger than memory units projected, the importance of memory units is significantly increased:

> *IF Memory_Sold – Memory_Projected IS Large THEN Importance_of_Memory = Increase_Significantly*

Such rule, when invoked would change the preference structure for the negotiator agents, which then would be willing to give up less memory units as a trade-off compared to other issues. This level could be termed "planning" layer. The manager agent would base its decisions solely on the information available to it, i.e. reports received from the sensors and negotiators. However, in reality, there are many other sources and types of information that may not be accessible to the agent. Furthermore, often judgmental input would be required to set the limits of authority for the manager agent, set the objectives and constraints (e.g. to maintain the ratio of failed vs. successful negotiations at a certain level) and attune its

parameters, e.g. its risk profile, or the thresholds for sensitivity for reacting to undesirable developments.
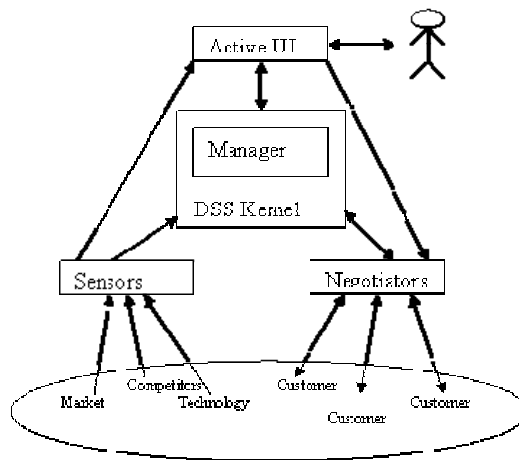

Figure 1. Decision Station for managing SLA negotiations

This is accomplished by the user through active user interface. Active user interface facilitates effective interaction with the user, while learning user preferences. This layer could be called the "judgmental" layer. Components of situated DSS could send various alerts to the user when human intervention may be desirable.

5. Prototype

To illustrate the benefits of the situated DSS approach and to obtain empirical results we have developed a prototype for conducting simulation experiments.

5.1 Buyer agents

The buyer agent is the initiator of the negotiation. It calculates a starting price for the complex service it wants to acquire resources for and sends the complete service level agreement to the seller. The buyer is able to accept the offer made by the seller agent or make a counteroffer. In the negotiation process the buyer agents and the seller agents only exchange service level agreements for one complex service.

For every resource the buyer agent has a price range ranging from the reservation level to the starting level which is equal to the acceptable level in the beginning. Out of these prices the total reservation, acceptable and starting price for a complex service can be calculated. To simulate different buyers we randomly generate these prices evenly distributed around a base price.

The buyer agent tries to negotiate in parallel with a specific number of seller agents. One negotiation step consists of an offer made by the buyer agent and the answer of the seller agent. The buyer agent negotiates only up to a specific number of steps with the seller agent

and abandons it after the limit and no agreement is reached. The buyer agent also abandons the seller agent if the seller agent aborts the negotiation or if the seller agent's offer is higher than reservation price. Every time a seller agent is abandoned the buyer agent tries to find a new one to keep negotiating in parallel with the same number of seller agents but only if the total number of allowed negotiation partners is not exceeded. An agreement is reached if the sellers offer is less or equal to the acceptable price of the seller and the negotiating cycle is complete and the buyer agent raises its price ranges. If no agreement can be reached in one step the buyer agent raises its acceptable prices. If the maximum number of negotiation partners is reached and no agreement could be made the negotiation cycle is complete and the buyer agent lowers its price ranges. As long as the buyer still has demand it tries to negotiate.

5.2 Decision Station for Sellers

The DS for the sellers agent incorporates the negotiating agents, the supervising agent, and the human interface. These three parts work together to maximize the seller's revenue in light of market uncertainties.

5.3 Negotiating Agent

Similar to the buyer agent the negotiating agent also has a price range for every resource which enables it to calculate the total price for a complex service. But the negotiating agent only lowers the price within its ranges when no agreement is reached. This part of the seller agent negotiates directly with the buyer agent. For every single buyer a separate negotiating agent is deployed.

In the negotiating process the negotiating agent checks if the buyer has accepted a previous offer. If this is the case the service level agreement for this complex service is agreed upon. In any other case the negotiating agent checks if the available resources can fulfill the buyer's complex service. If there are not enough resources available the negotiating agent aborts the negotiation. This is also the case if the buyer's price is below the reservation level. Similar to the buyer agent the negotiating agent accepts the offer if the price exceeds the acceptable price. If no agreement can be reached the negotiating agent lowers the acceptable price for the complex service and makes a counteroffer.

5.3.1 Supervising agent

After every negotiation cycle the supervising agent collects information from the negotiating agents, including how many agents are engaged in an active negotiation and how many have reached an agreement. The ratio of these two numbers is an indicator for the supervising agent whether the prices should be raised or lowered. The supervising agent also considers the percentage of each resource left to adjust their prices. The supervising agent can adjust the ranges of the negotiating agent within user specified boundaries. If these boundaries are reached the supervising agent advises the human decision maker to make a decision.

5.3.2 Human decision maker

The overall process is controlled by the human decision maker who is responsible for the initial settings. For example the base price for each resource, the ranges and adjustment rate for the negotiating agent and the boundaries and adjustment rate for the supervising agent. Figure 2 shows the user interface which allows the user to monitor the negotiation process and enables him to make decisions.

6. Simulations

In our simulations we set up 1000 buyers and 100 sellers each represented by an agent. To reflect the disparity in the demand and the supply the sellers possess by the factor of ten more resources compared to a typical demand by the buyers. The resources consist out of CPU hours, storage space and bandwidth. Every buyer has a demand for CPU hours and storage space which is broken down into several different complex services which also exhibit a certain amount of bandwidth. These complex services represent a typical application that can be outsourced on the Grid. The sellers possess a supply of the three resource types mentioned above.
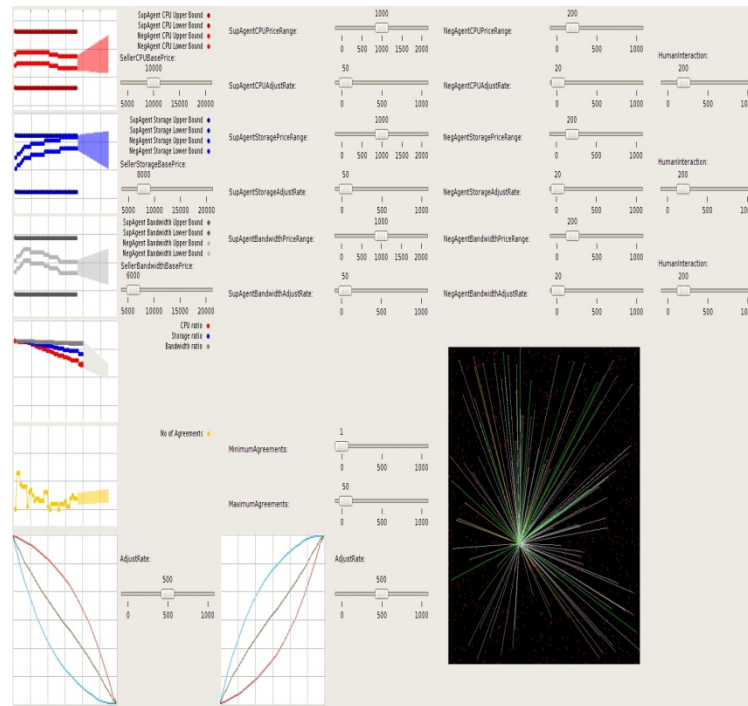

Figure 2. User interface

We have applied six different setting to our simulation setup. Mainly these are combinations of whether the buyer or the seller agents make concession or are fixed to their initial price levels. Additionally we have analyzed the performance of the Decision Station without any human interaction. According to these settings, we collected information about the total revenue and the revenue per resource generated by the seller agents. We also considered the remaining resources and the agreements reached in every step to determine the time needed to reach all possible agreements.

Figure 3 shows the total revenue for all seller agents. It is clearly visible that the simulations involving buyer-seller negotiations with (occasional) human interactions ended up with more total revenue for the seller agent than the ones without. Another interesting aspect is that even if the buyers do not make any concession it is better for the seller agent to make concession to reach his goal to maximize the total revenue.

The average revenue per sold resource is another examined criterion to determine the

effectiveness of the situated decision support system. Figure 4 indicates that the seller can make the most revenue per resource if he and the buyer make concessions. Without a human interaction this value shrinks. Figure 5 shows the remaining resources for the sellers. As it can be seen, the worst scenario is when none of the parties make concessions, i.e. fixed price case. Figure 6 shows the dynamics of the number of agreements reached. Even in the case where buyer does not make any concessions, sellers can benefit from dynamically adapting their offers in order to sell the remaining resources.

7. Conclusions

The present work proposed applying decision station approach to managing automated SLA negotiations. The framework is based on the model for situated decision support that effectively combines human judgment and autonomous decision making and action by agent components. The key idea behind the approach lies in the managing the fleet of negotiating agents by the use of a "manager" agent and human decision maker. We have illustrated the approach through simulation experiments.

The findings suggest that the decision station approach is superior to the fixed-price based mechanisms in terms of total revenues and efficient utilization of the available resources. It is also preferable to auction-based approaches which are complicated by the fact that the allocation problem is NP-complete, which causes serious problems with the growing number of resource providers and consumers. The approach presented in the paper promotes scalability while minimizing human, as well as computational effort.

Possible future work could be directed towards extending the situated DSS to workflow cases, where the service consumer engages into concurrent negotiations with many service providers such that SLAs will be made with respect to all constituents of the workflow. In addition, future work will comprise empirical testing involving human subjects.
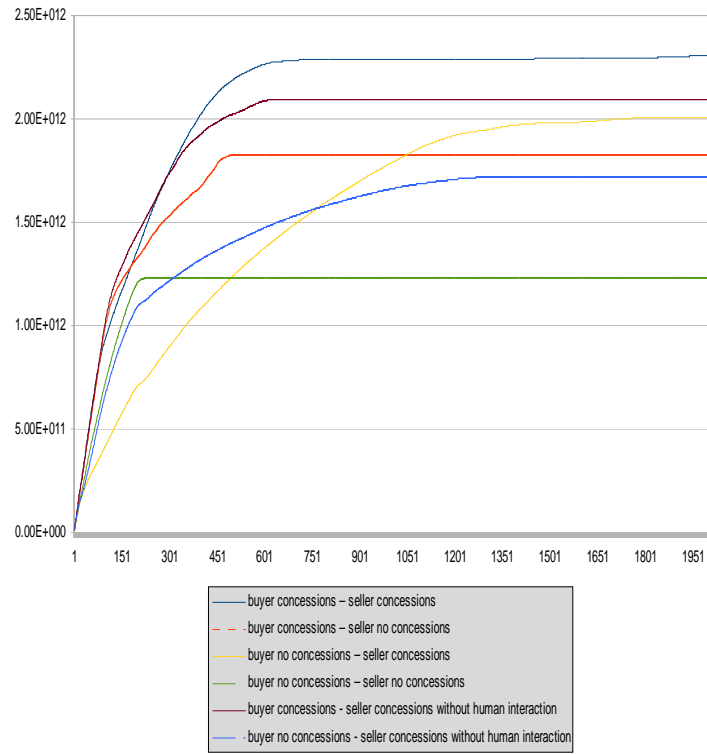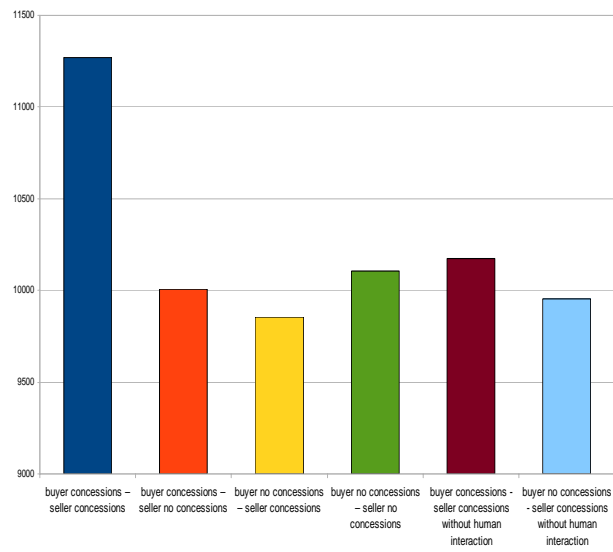
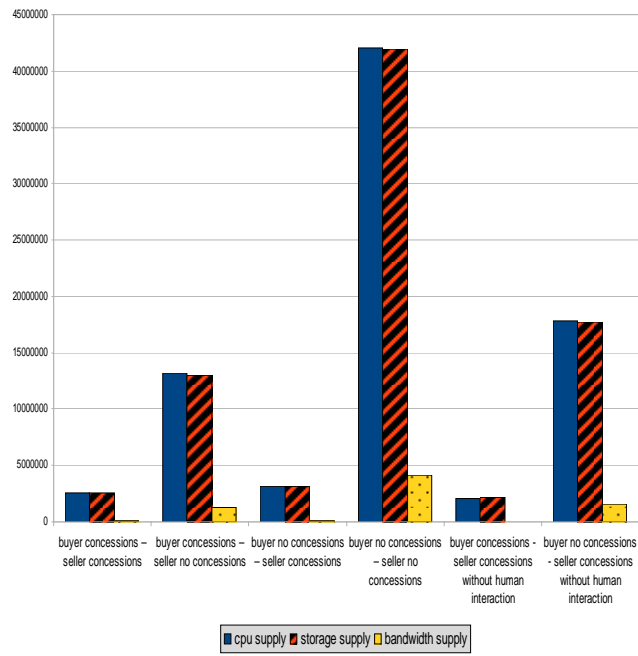**Figure 3. Total Revenues**

Figure 4 Revenue per resource

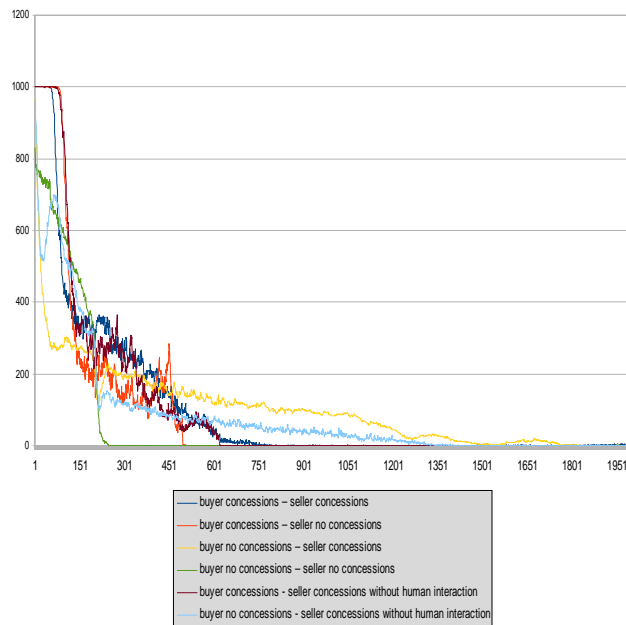**Figure 5 Remaining resources**

Figure 6 Number of agreements

References

Angehrn, A. A. and S. Dutta (1998). "Case-Based Decision Support." Communications of the ACM **41**(5): 77-86.

Carr, N. (2005). "The End of Corporate Computing." MIT Sloan Management Review **46**(3): 66-73.

Chen, E., R. Vahidov, et al. (2005). "Agent-Supported Negotiations in the e-Marketplace." Int. J. Electronic Business **3**(1): 28-49.

Czajkowski, K., I. Foster, et al. (2002). SNAP: A Protocol for Negotiating Service Level Agreements and Coordinating Resource Management in Distributed Systems. 8th Workshop on Job Scheduling Strategies for Parallel Processing, Edinburgh, Scotland.

Dang, J. and M. N. Huhns (2006). "Concurrent Multiple-Issue Negotiation for Internet-Based Services." Internet Computing **10**(6): 42-49.

Deora, V. and O. Rana (2007). "SLA Enforcement in Grid using Runtime Violation Prediction and Reaction." Working Paper.

Eymann, T., D. Neumann, et al. (2006). On the Design of a Two-Tiered Grid Market Structure. Business Applications of P2P and Grid Computing, MKWI 2006.

Faratin, P., C. Sierra, et al. (2002). "Using Similarity Criteria to Make Issue Trade-Offs in Automated Negotiations." Artificial Intelligence **142**: 205-237.

Gimpel, H., H. Ludwig, et al. (2003). PANDA: Specifying Policies for Automated Negotiations of Service Contracts. 1st International Conference on Service Oriented Computing (ICSOC 03), Trento, Italy.

Hasselmeyer, P., C. Qu, et al. (2006). Towards Autonomous Brokered SLA Negotiation. eChallenges 2006, Barcelona.

Hess, T. J., L. P. Rees, et al. (2000). "Using Autonomous Software Agents to Create Next Generation of Decision Support Systems." Decision Sciences **31**(1): 1-31.

Jarke, M., M. T. Jelassi, et al. (1987). "MEDIATOR: Towards a Negotiation Support System." European Journal of Operational Research **31**(3): 314-334.

Kersten, G. E. and G. Lo (2003). "Aspire: Integration of Negotiation Support System and Software Agents for E-Business Negotiation." International Journal of Internet and Enterprise Management (IJIEM) **1**(3): 293 - 315.

Li, H., S. Y. W. Su, et al. (2006). "On Automated e-Business Negotiations: Goal, Policy, Strategy, and Plans of Decision and Action." Journal of Organizational Computing and Electronic Commerce **13**: 1-29.

Ludwig, H., H. Gimpel, et al. (2005). Template-Based Automated Service Provisioning - Supporting the Agreement-Driven Service Life-Cycle. International Conference on Service Oriented Computing.

Nguyen, T. D. and N. R. Jennings (2004). Coordinating multiple concurrent negotiations. 3rd Int. Conf. on Autonomous Agents and Multi-Agent Systems, New York, NY.

Ouelhadj, D., J. Garibaldi, et al. (2005). A Multi-agent infrastructure and a Service Level agreement Negotiation Protocol for Robust Scheduling in Grid Computing. Advances in Grid Computing - EGC 2005. T. P. A. R. Peter M.A. Sloot Alfons G. Hoekstra, Marian Bubak. Heidelberg, Springer**:** 651-660.

Parkin, M., D. Kuo, et al. (2006). A Framework & Negotiation Protocol for Service Contracts. Proceedings of the IEEE International Conference on Services Computing.

Rahwan, I., R. Kowalczyk, et al. (2002). "Intelligent Agents for Automated One-to-Many e-Commerce Negotiation." Australian Computer Science Communications **24**(197-204).

Rana, O. (2006). "OGF-19 Report." <u>Technical Report</u>.

Rao, H. R., R. Sridhar, et al. (1994). "An Active Intelligent Decision Support System." <u>Decision Support Systems</u> **12**(1): 79-91.

Sahai, A., S. Graupner, et al. (2003). <u>Specifying and Monitoring Guarantees in Commercial Grids through SLA</u>. Conference of Cluster Computing and Grids (CCGrid).

Siddiqui, M., A. Villazon, et al. (2006). <u>Grid Capacity Planning with Negotiation-based Advance Reservation for Optimized QoS</u>. Proceedings of the ACM/IEEE Conference on Supercomputing, Tampa, USA.

Vahidov, R. (2002). <u>Decision station: a Notion for a Situated DSS</u>. 35th Hawaii International Conference on System Sciences.

Vahidov, R. and B. Fazlollahi (2004). "Pluralistic Multi-Agent Decision Support System: A Framework and an Empirical Test." <u>Information & Management</u> **41**(7): 883-398.

Vahidov, R. and G. E. Kersten (2004). "Decision Station: Situating Decision Support Systems." <u>Decision Support Systems</u> **38**(2): 283-303.

Van de Walle, B., S. Heitsch, et al. (2001). <u>Coping with One-to-Many Multi-criteria Negotiations in Electronic Markets</u>. 12th International Workshop on Database and Expert Systems Applications.