# A Non-Additive Negotiation Model for Utility Computing Markets

Mario Macías and Jordi Guitart[1] [2]

*Abstract*— **Market-based resource allocation is a promising model for dealing with the growing Utility Computing environments, such as Grid or Cloud Computing. Agents that represent both service clients and providers meet in a market to negotiate the terms of the sale of resources. Additive negotiation models are extended because they are simple, but they are not valid for negotiations whose terms are not independent between them. This paper proposes a simple non-additive model for performing negotiations and demonstrates its validity through simulation experiments.**

*Key words*— **Utility Computing, Grid Computing, Cloud Computing, SLA Negotiation, nonadditive utility functions**

## I. Introduction

In the recent years, the big mainframes paradigm where users own their computing resources[1], is being progressively transiting to a more utility-driven paradigm, where users do not own their resources and pay for the usage of remote resources [2].

The Grid[3] and, more recently, Cloud Computing[4] are the most promising current implementations of Utility Computing, the first in scientific and academic environments, and the second in the business world. This new evolution has made the classical Resource Management mechanisms very inefficient because some reasons, such as the growing complexity of finding optimum resource allocations, conflicts of interest between users from different organisations, introduction of new business metrics[5], worldwide dispersion of the systems, etc.

Having into account these arguments, large systems seem to be too complex to be managed centrally. Market-based resource management is proposed as a decentralised paradigm to deal with the complexity by the next reasons:

- The possibility of doing business will motivate service providers to offer their resources in the system and give a Quality of Service (QoS) according to their real capacity.
- Market mechanisms obligate the users to adjust their reservations of the system, both in time and space, to their real requirements.
- It is relatively easy to implement in a decentralised architecture.
- The complexity is reduced, because participants enter in the market looking for the satisfaction of their own necessities, and they do not need to know the global status of the system to maximise their utility.

[1]Barcelona Supercomputing Center, Jordi Girona 29, 08034 Barcelona, Spain, e-mail `mario.macias@bsc.es`, `jordi.guitart@bsc.es`

[2]Technical University of Catalonia, Jordi Girona s/n, 08034 Barcelona, Spain, e-mail `mario@ac.upc.edu`, `jguitart@ac.upc.edu`

In market-based utility computing scenario, brokers that represent service providers or clients participate in a market to sell or buy their services. When the clients find there their necessities, a **negotiation process** is started to establish the terms of the contract (QoS, price, time slots). If both parts reach an agreement, the terms of the contract are specified in a Service Level Agreement (SLA) and the client application can acquire the bought resource. During the usage of the resources, the component called SLA Enforcement watches for the correct fulfilment of the terms of the SLA, and penalises the buyers or the sellers if they violate the SLAs.

Since brokers that negotiate for the buying and selling of services are autonomous agents (this is, they communicate between them and take decisions without human intervention), it is needed to provide them some economic models and intelligent behaviour so they are able to take the best decisions for their represented actors in the market (client applications or service providers) and maximise their utility.

This paper enhances existing economic models for negotiation and applies them to the sales of services and resources between computing agents: when the resource broker negotiates an SLA with the broker of a Client, it has into account some economic terms, such as price, penalties for contract violation, time, etc. In addition, there are other terms in the SLA, which are more technical, and also can have influence in the economic terms, specially those related with the Quality of Service (QoS) (e.g. throughput, response time...) or those related with the sales of plain resources (number of CPUs, speed, memory...). For a purely-economic resource broker, it is very difficult to quantify the terms of the SLAs, since it has not enough technical knowledge about the status and punctual capacities of the resources, since it determines if a task can be executed or not, and the minimum price to make this task profitable for the resource provider.

Having these arguments into account, the main contributions of this work are:

1. Modelling and characterisation of the negotiations required to perform sophisticated sales in Market-Based Utility Computing in function of the desirable objectives, that also will be defined and studied.
2. Evaluation of the proposed economic models for the negotiation between brokers for the sale of

computing utilities. This includes the comparison of several values for the parameters of the model and the evaluation about its feasibility and influence in the achievement of desired objectives.
3. Usage of low-level dynamic knowledge, provided by the resource fabrics, for giving support to economic negotiations. The required knowledge is defined by the contributions enumerated in point 1 and 2, and is acquired in real-time from the monitorisation of the Resource Fabrics.

The rest of the paper is organised as follows. Section II describes the component created to deal with both economic and technical issues in the market. Section III describes the scenario where the negotiations carry out. Section IV described the models used for the negotiation. Section V describes the evaluation environment and the results of the simulation. Finally, section VI shows the conclusions of this work and points to some future research lines.

## II. Economically Enhanced Resource Management

To help market brokers taking decisions about the sales of the resource fabrics, an Economically Enhanced Resource Manager (EERM) [6], [7] is introduced as an intermediate point to deal with the economical and technical issues.

The EERM interacts with various other components, namely a Market Broker and the Resource Fabrics. There is a Utility Computing Market, which is a middleware where client and provider brokers query the prices and offer the services on the Market, respectively. Resource Fabrics refers to Grid Middlewares such as Condor [1], Globus [8], or Web Application Servers such as Tomcat.

The main functions of EERM are:

- Decide whether incoming tasks are accepted or not, based of the usage of the resources for a given time slot, the client priority, and the sale price.
- Calculate prices for offered jobs and services based in current market status, current resource usage and predictions about the impact of a job in the usage of resources.
- Check that the accepted SLAs can be kept. If one or more SLAs can not be fulfilled it takes the decision to suspend or cancel jobs to ensure the fulfilment of the other SLAs and maximise overall revenue.
- It is responsible of the communication with the local resource managers and influences the local resource management to achieve a more efficient global resource use.

The behaviours in the previously enumerated functions can be tuned with policies.

## III. Negotiation Protocol

Before the negotiation starts, the **EERM** of the Service Provider must register its offered services in the **Service Discovery component**. For each service, it is provided some semantic information that allows to identify what service is and its functionalities, and an extra meta-SLA with some data about the SLA terms (also known as Service Level Objectives) that the service provider is willing to negotiate, such as response time, throughput, quality (for example in video compression), duration of the service, etc.

When a **Client Broker** wants to acquire a service, it queries the Service Discovery by providing some semantic information, and the Service Discovery returns a list of the Service Providers that match the requirements (every provider has its own EERM) and the meta-data about the negotiable SLA terms.

The next process is similar to the one described in the WS-Agreement specification [9], but with an extra iteration in the negotiation. Before starting the negotiation the Client Broker selects the most interesting services, and creates a proposal of agreement for each one; using the meta-data it creates an uncompleted SLA with its requirements, and leaves other SLOs as void, to be completed by the EERM in the next negotiation process.

When the EERM receives the SLA proposal, it evaluates if the proposed terms can be accepted. If the Client broker received from the EERM an acceptance message or a counter-offer, it evaluates it and finishes the negotiation by not accepting the SLA or by sending a confirmation message to the EERM.

### A. Assumptions

Since this paper is focused in the negotiation process, some components of a Market are ignored and assumed that they are present:

- There is already a service discovery mechanism where providers can offer their services, and clients can look for them by providing some semantic information.
- The EERM has a SLA decomposition component that is able to translate from the Service Level Objectives (SLO) that compose the SLA, such as response time or throughput, to system-level thresholds, that are used to assign the correct number of resources to a service, for fulfilling the agreed SLA.

## IV. Negotiation Model

### A. Characterisation of the negotiation

Once the scenario has been defined, it is time to define the content and details of the research of this paper. Since it is focused in the usage of resource-level information for enhancing the SLA negotiation in utility computing markets, the first step to do is to characterise the negotiation. To do that, this paper takes into account the *Organizing Questions* that Raiffa proposed in his book [10].

*Two monolithic parties* Broker agents can take the negotiation decisions autonomically.

*Repetitive game* What one of the parties do in a negotiation can be taken into account in future negotiations (reputation)

*Multiple issues* Such as price, time and several SLOs.

*Some fixed threats* If there is no agreement, the client will look for another provider; if the provider oversells its resources, there is the threat breaking the SLAs.

*Binding contracts* And there are entities such as *SLA Enforcement* to check at any moment that the contracts are being fulfilled and, if not, make the violating party to pay a penalisation.

*Cooperative antagonism* The disputants recognise that they have differences of interests; they would like to find a compromise. They do not have malevolent intentions, but neither are they altruistically inclined.

### *B. On the usage of non-additive utility functions*

The first issue that must be defined is the analytic model for representing the negotiations that will be performed by the *Economy Agent*. This model must take into account the negotiated SLOs and other terms, such as client classification or reservation slots plus the sale price.

Traditional negotiation models for utility computing are based in the *"two parties, many issues"* model proposed by Raiffa[10] and Faratin [11]. This model is pretty easy to manage and calculate the maximum and minimum utilities: during the negotiation process, the provider can know easily which addends to increase or decrease in order to increase the utility and keep it the nearest possible to the maximum. However, it has a problem: it is an *additive model* which assumes that all the factors are independent from the others. This is not true, since some factors such as the price are strongly related with the SLOs, the type of client, etc.

Let $S$ be the SLA under negotiation, the **non-additive utility** function $U$ used in this paper for the service provider is the next:

$$U(S) = \sum_{i=1}^{m} o_i u_i(S) \tag{1}$$

Where $m$ is the number of goals for the provider, such as revenue maximisation, high reputation, performance maximisation, high occupation of resources, satisfaction of certain type of users, etc. $u_i$ is the sub-utility function that defines how much will be the objective $i$ satisfied, and $o_i$ is a number between 0 and 1 that defines the priority that the provider assigns to the concrete objective. It must be considered that $\sum_{i=1}^{m} o_i = 1$.

Although Equation 1 is similar to an additive function, actually it is not. Instead of calculating each of the sub-utility functions in function of a single SLA term and finally add them up, Equation 1 calculates all the sub-utilities in function of the whole SLA. This is because the different objectives are not independent from the others and, for example, revenue maximisation can affect negatively the client satisfaction. Following this example, an additive function will suppose that increasing the revenue will always increase the general utility, but that might require to break some agreed SLAs [7], [12] and in consequence, decrease the general user satisfaction. If the weight assigned to the user-satisfaction goal is higher than the assigned to revenue maximisation, the global utility will decrease instead of increase.

The usage of non-additive utility functions will allow a more accurate definition of the utility; however, finding the maximum utility analytically is much more difficult.

### *C. Negotiation terms and Utility functions*

First is needed to define the set $O$ of objectives, the set $S$ of SLA terms and the utility function $U(S)$ that calculates how beneficial the proposed SLA for the objectives of the provider is.

Let $O \subseteq \{o_{rv}, o_{cc}, o_{ph}, o_{rp}\}$ the set of objectives, where:

- $o_{rv}$ is the objective that defines the maximisation of the revenue. The higher is the revenue the higher is $u_{rv}$.
- $o_{cc}$ is an objective used for client classification [13]. This gives preference to the local users (or users from a near organisation) over the users from non-related organisations.
- $o_{ph}$ is the objective that gives preference to tasks or services to be executed in off-peak hours, to prevent the system overload during peak hours.
- $o_{rp}$ is the objective used for maximising the reputation of the provider [14].

The simulations performed in this paper demonstrate how the behaviour of the provider can be modulated only by changing the values of the components of $O$ that multiply their associated sub-utility functions $u_{rv}, u_{cc}, u_{ph}, u_{rp}$ in negotiation time as can be shown in the general utility function (Equation 2).

$$U(s) = o_{rv}u_{rv} + o_{cc}u_{cc} + o_{ph}u_{ph} + o_{rp}u_{rp} \tag{2}$$

The next subsections describe and justify the sub-utility functions in the simulation, calculated in base to the SLA $S \subseteq \{M, C, CP, Rev, \Delta t\}$, where $M, C$ are the Memory and CPUs amount to acquire, $0 \leq CP \leq 0$ is the indicator of Client Priority, $\Delta t$ is the time slot where the resources are assigned and $Rev$ is the revenue acquired by the sale. All the sub-utilities are normalised to the same range $[-1, 1]$ because otherwise the influence of the weights $O$ would be distorted by the differences between the ranges of the sub-utilities.

#### C.1 Revenue Maximisation

When negotiating the sale of a good, the **reservation price** of the seller is the minimum price that the seller can accept without losing money for the sale. The **reservation price** of the buyer is the maximum

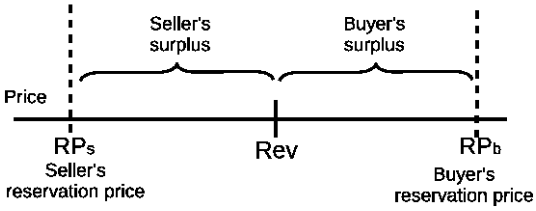


Fig. 1. Reservation Price of both Buyer and Seller

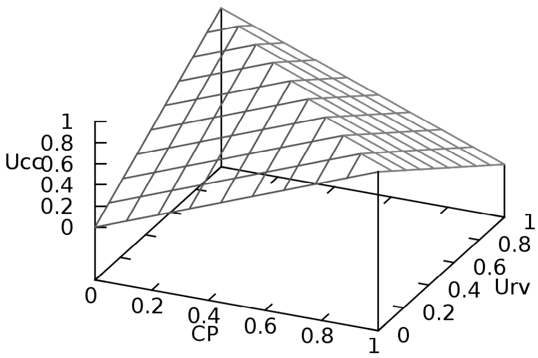


Fig. 2. Utility function used for Client Classification

price it can pay and do a sale which is beneficial for its objective (see figure 1).

Equation 3 defines the utility for given revenue:

$$u_{rv}(S) = \frac{Rev - RP_s}{RP_b - RP_s} \tag{3}$$

The main issues of implementing this formula are to know the reservation price of the buyer, which only can be speculated. In the simulations performed, it has been defined an **aggressiveness factor**, that increases or decreases $RP_b$ in function of the status of the resources: too much free resources imply that the prices are too high in a competitive market, so $RP_s$ is being progressively reduced to match the equilibrium status in sales.

C.2 Client Classification

In this paper, client classification is performed through price discrimination [13]. The parameter $CP$ is the Client Priority, it tends to 1 when the Client is much related to the organisation of the provider, and tends to 0 when there is absolutely no relation between the Client and the Provider.

Equation 4 is used to define the utility for client classification. Given $u_{rv}$ and $CP$, if the Client priority is high, the utility will be higher when $u_{rv}$ is low (the provider must not be expensive for related clients). If the Client priority is low, the utility will be higher when $u_{rv}$ is high (see figure 2).

$$u_{cc}(S) = \begin{cases} CP + u_{rv} & \text{if } u_{rv} < 1 - CP \\ 2 - CP - u_{rv} & \text{otherwise} \end{cases} \tag{4}$$

C.3 Priorisation of Off-Peak hours

Let $\Delta t = t_f - t_i$ be the interval of time where the task is executed, $C_{tot}(t)$ be a constant function whose value is the number of CPUs of the Provider, $C(t)$ be a constant function whose value the number of CPUs requested to the task under negotiation, and $C_{used}(t)$ be a function that describes the number of busy CPUs in the provider over time. Equation 5 is the utility function that is higher when more resources are free, and near 0 when the provider resources are near its maximum occupation.

$$u_{ph}(S) = 1 - \frac{\int_{t_i}^{t_f} C_{used}(t) + C(t)\,dt}{\int_{t_i}^{t_f} C_{tot}(t)\,dt} \tag{5}$$

In the simulation, both CPU and Memory are negotiated. But since CPU is the bottleneck, it is used as a resource for calculating the peak hours.

C.4 Utility for reputation

Let $R_0$ be the reputation of the provider in negotiation time, $R$ be the future reputation of the provider in case of SLA violation (updated as described in [14]), and $P$ the probability of violating an SLA (calculated in base to past statistical data); Equation 6 shows the utility of keeping the provider's reputation:

$$u_{rp}(S) = \frac{P\,R + (1-P)R_0}{R_0} - 1 \tag{6}$$

Notice that this utility function ranges from -1 to 0, since losing the current reputation is bad, and keeping it is neither good nor bad for the provider.

## V. Evaluation

This section describes the experiments performed for checking the validity of the model. A statistical method is used to check the influence of each of the parts of the model. Due to resources limitations (hundreds of different providers are compared in a same market), the model has been checked using a simulation of a market.

### *A. Simulation Environment*

A simple market has been simulated to test the validity of the negotiation model. A Client Broker that can represent a Web Client or a Grid Client enters in the market to ask for web workload or for plain resources. The workload for grid has a pseudo-random distribution and the workload for web services follows a distribution as shown in figure 3, taken from a real web application, with peak and off-peak hours.

Grid Clients send a SLA proposal where is specified the plain resources (CPU and Memory) to buy, the duration of the job, and a time interval where the job can be executed (bigger than the duration, to let the EERM schedule the best execution time). Web Service Clients send a required workload for a service, and a fixed time interval to use the services (there is no arbitrary schedule of the reservation, since web users want the services for the same moment).

A Client looks for potential providers in the market discovery and sends SLA Proposals to all of them.

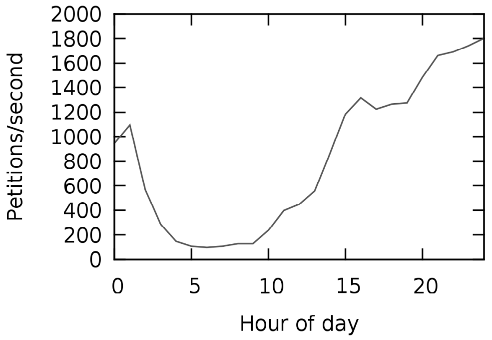


Fig. 3. Sample of Daily workload of a Web Server

| | Dem. exc. | Off. exc. | Equil. |
|---|---|---|---|
| Avg. Price | 0.487 | -0.402 | -0.452 |
| Revenue | -0.542 | -0.410 | -0.454 |

TABLE I

CORRELATIONS BETWEEN $o_{rv}$ AND THE AVERAGE PRICE PER CPU USED AND THE TOTAL REVENUE

After that, the providers accept/deny the proposals and return to it a time allocation and a price, based on the maximisation of their utility function. Finally, the Client chooses the Provider with a best price and time schedule for its interests and sends it a confirmation.

The provider can violate the SLA due to an internal error, or because it receives a proposal from another Client that can not be allocated but is interesting to accept it and cancel the other (it is decided by the utility function having into account objectives such as client classification or revenue maximisation) [12], [7] . This violation will affect to the reputation of the provider, which is taken into account by the Client in negotiation time.

The simulation is performed in a competitive market with 100 providers, whose objective weights of the utility function are generated randomly.

## *B. Simulation Results*

This section shows the results of the simulation in terms of the four objectives described in section IV-C.

### B.1 Revenue Maximisation

Table I shows how the influence of $o_{rv}$ in the average price per CPU/hour and the total revenue of the provider in function to the market status: demand excess, offer excess, and equilibrium [15].

In an Demand Excess scenario, where the provider can be more aggressive in its negotiations (because there are more clients interested on acquiring its services), it can be observed a positive correlation between $o_{rv}$ and the average price per CPU/hour. However the correlation with the total revenue is negative, because higher prices will entail clients to look for other providers and there are fewer sales of resources. When there is Offer Excess or the market is in Equilibrium, $o_{rv}$ has a negative influence both in the revenue and in the average price, because the sales are very low and the aggressiveness coefficient defined in IV-C.1 decreases dramatically.

| | Dem. exc. | Off. exc. | Equil. |
|---|---|---|---|
| Avg. Price | -0.504 | -0.166 | -0.199 |
| Revenue | 0.280 | -0.143 | -0.130 |
| Avg. Affinity | 0.603 | 0.249 | 0.314 |

TABLE II

CORRELATIONS BETWEEN $o_{cc}$ AND THE AVERAGE PRICE PER CPU, THE TOTAL REVENUE AND THE AVERAGE AFFINITY OF CLIENTS

| | Dem. exc. | Off. exc. | Equil. |
|---|---|---|---|
| Avg. Price | 0.292 | 0.471 | 0.586 |
| Revenue | 0.243 | 0.451 | 0.527 |

TABLE III

CORRELATIONS BETWEEN $o_{ph}$ AND THE AVERAGE PRICE PER CPU USED AND THE TOTAL REVENUE

### B.2 Client Classification

Table II shows the effectiveness of the inclusion of $o_{cc}u_{cc}(S)$ in the general utility function: the higher is $o_{cc}$, the higher is the affinity. This can be specially observed when there is a demand excess and all the prices grow dramatically: the providers with high affinity are a refuge for the clients, where they can get better prices.

As described in section IV-C.2, $u_{cc}$ is strongly related with $u_{rv}$. Table II reflects this relation as a negative correlation between the $o_{cc}$ and the average price. However, related to the total revenue, the correlation is positive in the demand excess scenario because providers with low prices sell more resources.

### B.3 Priorisation of off-peak hours

Workload for Web Services have fixed intervals but an irregular distribution (figure 3), and workload for Grid Tasks have a random distribution, but since they are not real-time applications, they can be scheduled to be executed in the future. Figure 4 shows how the inclusion of $o_{ph}u_{ph}(S)$ in $U(S)$ makes the providers the possibility for giving better prices to the clients in off-peak hours and, in consequence, the Grid Jobs are automatically executed when the Web Services workload is low.

Table III shows how $o_{ph}$ has a positive influence on the money earned by the provider. This is because the higher is $o_{ph}$, more jobs are allocated in the off-peak hours and the provider earns in this time slot a money that otherwise would not earn, and it also makes lower the price of peak hours, making the resources more attractive for web-services clients.

### B.4 Reputation

Figure 5 shows clearly the importance of keeping a high reputation. In the experiments, the revenue increases almost linearly with the reputation. At equal

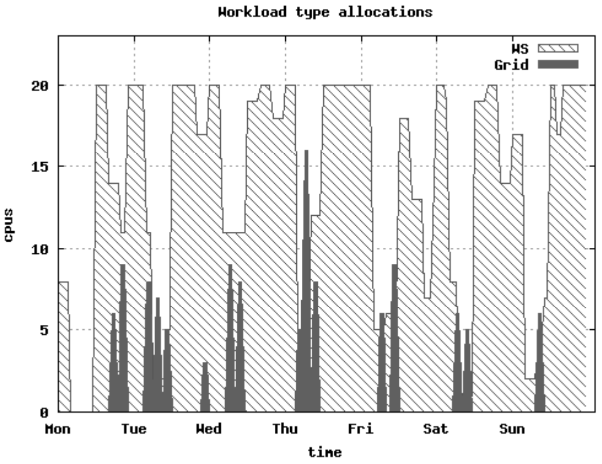


Fig. 4. Allocation in time of workloads divided by Web Services or Grid

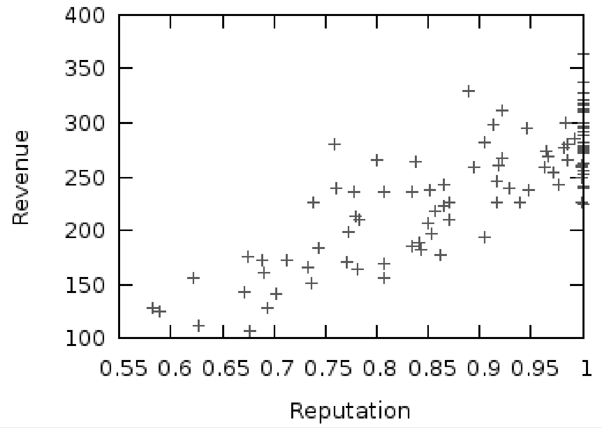


Fig. 5. Relation between reputation and revenue

prices, a Client will choose the Provider with higher reputation. The alternative to providers with low reputation is to decrease their prices.

## VI. Conclusions and future work

This paper intends to be a step forward in the modelling and evaluation of utility functions for negotiations in utility computing markets. The simulations show how a provider can perform complex actions by only maximising a multi-dimensional utility function. The contribution of these experiments lies in the usage of non-additive utility functions, more difficult to maximise, but assume that the terms under negotiation are not independent between them.

The proposed non-additive utility function considers the possibility of having multiple objectives in a same entity, such as revenue maximisation, client classification, reputation or load-balancing in time. In order to keep the efficiency both in economic and performance terms, most of the parameters that compose the utility function are collected dynamically from the resource-level information. The simulations performed demonstrate how the objectives can be partially achieved by balancing correctly their weights in the utility function.

This work leaves open some lines for future research: the first is to improve and extend utility functions and to work on better methods for their maximisation; other future work is to evaluate the validity of the model in real environments, not simulations, taking real data from the resource fabrics, and compare it with other existing models.

## Acknowledgement

This work is supported by the Ministry of Science and Technology of Spain and the European Union (FEDER funds) under contract TIN2007-60625 and Commission of the European Communities under IST contract 034286 (SORMA).

## Referencias

[1] J. Basney and M. Livny, "Deploying a high throughput computing cluster," in *High Performance Cluster Computing: Architectures and Systems, Volume 1*, R. Buyya, Ed. Prentice Hall PTR, 1999.

[2] M. A. Rappa, "The utility business model and the future of computing services," *IBM Syst. J.*, vol. 43, no. 1, pp. 32–42, 2004.

[3] D. Thain and M. Livny, "Building reliable clients and servers," in *The Grid: Blueprint for a New Computing Infrastructure*, I. Foster and C. Kesselman, Eds. M. Kaufmann, 2003.

[4] R. Buyya, C.S. Yeo, and S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities," *High Performance Computing and Communications, 10th IEEE International Conference on*, vol. 0, pp. 5–13, 2008.

[5] A. Van Moorsel, "Metrics for the internet age: Quality of experience and quality of business," Tech. Rep., 5th Performability Workshop, 2001.

[6] T. Püschel, N. Borissov, M. Macías, D. Neumann, J. Guitart, and J. Torres, "Economically enhanced resource management for internet service utilities.," in *WISE*, B. Benatallah, F. Casati, D. Georgakopoulos, C. Bartolini, and W. Sadiq, Eds. 2007, vol. 4831 of *Lecture Notes in Computer Science*, pp. 335–348, Springer.

[7] M. Macías, O. Rana, G. Smith, J. Guitart, and J. Torres, "Maximizing revenue in Grid markets using an Economically Enhanced Resource Manager," *Concurrency and Computation: Practice and Experience*, p. n/a, September 2008.

[8] Ian T. Foster, "Globus toolkit version 4: Software for service-oriented systems.," in *NPC*, Hai Jin, Daniel A. Reed, and Wenbin Jiang, Eds. 2005, vol. 3779 of *Lecture Notes in Computer Science*, pp. 2–13, Springer.

[9] "Web Services Agreement specification (WS-Agreement)," .

[10] H. Raiffa, *The art and science of negotiation*, Belknap Press of Harvard University Press, Cambridge, Mass, 1982.

[11] P. Faratin, C. Sierra, and N.R. Jennings, "Negotiation decision functions for autonomous agents," *International Journal of Robotics and Autonomous Systems*, vol. 24, pp. 3–4, 1998.

[12] M. Macías, G. Smith, O. Rana, J. Guitart, and J. Torres, "Enforcing service level agreements using an economically enhanced resource manager," in *Proceedings of hte 1st Workshop on Economic Models and Algorithms for Grid Systems (EMAGS 2007)*, Austin, Texas, USA, September 2007.

[13] T. Püschel, N. Borissov, D. Neumann, M. Macías, J. Guitart, and J. Torres, "Extended resource management using client classification and economic enhancements," in *Proceedings of eChallenges e-2007 Conference*, The Hague, Netherlands, October 2007.

[14] M. Macías and J. Guitart, "Influence of reputation in revenue of grid service providers," in *Proceedings of the 2nd International Workshop on High Performance Grid Middleware (HiPerGRID 2008)*, Bucharest, Romania, November 2008.

[15] J. Guitart, M. Macías, O. Rana, P. Wieder, R. Yahyapour, and W. Ziegler, *Market-Oriented Grid and Utility Computing*, chapter SLA-based Resource Management and Allocation, Number 12. Wiley, 2009.