Engineering Grid Markets

Dirk Neumann

Information Management and Systems, University of Karlsruhe (TH), Germany
neumann@iism.uni-karlsruhe.de

1. Introduction

Grids denote a promising concept to pool computer resources for joint computations. Facing increasingly more complex and demanding resources, Grids are deemed the solution to those problems by a more efficient and flexible usage of already existing resources. From a technical perspective Grid middleware have made significant progress. While in former implementations it was only possible to share idle resources (e.g. using Condor), new Grid middleware allow advance reservation of resources that are once committed not usable locally for the committed time (e.g. GRAM in Globus Toolkit 4.0). Advance reservation thus allows the sharing of not only idle resources but of all designated resources.

From an economic point of view, Grids that are relying on the sharing principle will flinch for the same reasons as P2P systems did. The reason stems from the lack of incentives on the resource provider side. Contributing resources to the Grid is associated with costs but no benefits. As such selfish resource owners refrain from contributing but not from consuming. As Hardin's *"tragedy of commons"* predicts, such a system where all consume but nobody's contributing will eventually collapse. This development can be seen even in the e-Science community, where trust and reputation among the community members is recognized.

Since the 60s researchers have motivated the use of markets as a means to cope with those incentival problems in distributed computing. The first attempt has been made with auctioning off time slots of the Harvard supercomputer (Sutherland 1968). While this primer was purely paper-based and restricted to one single computer, subsequent proposals and prototypes offered automated trading in distributed environments. Despite the fact that the idea of using markets in distributed computing is not new, no implementation has made it into practice. There are several reasons for this unsuccessful development, stemming from limitations (1) on the operating system level (e.g. inadequate distributed identity management and lack of kernel-supported resource isolation) and (2) on the level of market design (e.g. insufficiently rich bidding languages and clearing as well as pricing policies).

In the meantime significant progresses have been made on the operating system level, upfront in the virtualization infrastructures such as Xen Hypervisor or VMWare Server, removing more and more the technical obstacles. In unison, significant advancements have been noted in the area of combinatorial auctions. Apparently, the foundation for establishing vivid markets has been set. In this paper, those new developments will be considered while designing a market for Grid. Experience has shown that engineering markets is everything but easy, due to the fact that markets are very sensitive to the underlying economic environment (i.e. what is being traded, who are the participants, what is the technical infrastructure etc.). Thus, this paper adopts a Market Engineering approach and follows systematically through the phases of the process model (Weinhardt et al. 2003).

The contribution of this paper is threefold. Firstly, this paper derives a requirement list stemming from Grid applications that need to be fulfilled by the market-based Grid. Secondly, the paper compares related work with the above requirements. Thirdly, and most importantly, this

Dagstuhl Seminar Proceedings 06461
Negotiation and Market Engineering
http://drops.dagstuhl.de/opus/volltexte/2007/1004

paper provides a fully-fledged market mechanism that is tailored to the use in service-oriented Grids.

The remainder of this work is structured as follows. In section 2 the market engineering process model is briefly shown as it prescribes the organization of this paper. In section 3 the requirements of service-oriented Grid applications are deduced. As the requirements strongly depend on the specific application, we present a concrete application and use it as motivating scenario. Related work is compared with respect to those requirements. In section 4 a lightweight market mechanism is tailored specifically for the motivating scenario. Section 5 concludes with a summary and an outlook on future work.

2. Market Engineering

Engineering an adequate market that satisfies the requirements stemming from the environment is a complex task involving several different design activities. Any design activity can thereby be intuitive or systematic:

- **Intuitive design activity**

  An intuitive design activity relies on creativity in the form of complex associations of ideas. Methods that attempt to foster intuitive design activities at increasing the flow of ideas (e.g. Delphi-method, brainstorming). The drawback of intuitive activities is that good ideas are not discovered or undiscovered – they are unpredictably developed. In addition, the results of intuitive activities are strongly dependent on the designer’s expertise, skills, and experiences. Even worse intuitive ideas are often already circumscribed by the education and experience of the designer. The most severe problem for intuitive design activities is that they frequently fail in complex situations.

- **Systematic design activities**

  It is likely that not a single design activity solves the entire design problem, but just parts of it. Thus, a strategy is needed which decomposes the complex overall problem into several smaller – ideally less complex – problems. This strategy suggests that problems are not approached in their totality; instead the complex problem is transformed into smaller problems, for which stronger design methods, but also intuition may exist that solve them. Hence, systematic design activities apply deliberate, step-by-step procedure to aid the designer in the matching of the unique problem situation along the overall design process with the available design methods (Grant 1979).

In the context of designing markets for Grids, the approach of Market Engineering aims at the systematic, goal-oriented development of the market infrastructure. In essence, Market Engineering provides a process model that guides how design can be structured.

Figure 1 illustrates the higher-level phases of the Market Engineering process (Weinhardt et al. 2003). The beginning of the Market Engineering process is marked by the definition of objectives and the strategy that governs the Market Engineering approach. In the environmental analysis – first stage – the requirements of the new market mechanism are elicited. Typical questions that need to be addressed comprise for example what user groups may want to engage the market and what are their needs and constraints?

Subsequently follows the design and implementation stage, which is a container for several design phases. In analogy to the engineering design process from mechanical engineering (Pahl et al. 1984), the design stage is decomposed into four major phases being the conceptual design, embodiment design, detail design, and implementation.

For Market Engineering the conceptual design refers to the design of the market mechanism consisting of an allocation function and a payment scheme. These purely conceptual functions are refined in the embodiment design phase into an auction model which is afterwards transformed into a formal process model (e.g. using FSM). In the detail design phase, all remaining design issues are tackled and subsequently implemented. Having implemented the appropriate market mechanism, it is tested upon its economic properties and its operational functionality. A detailed description of the design phases can be found at (Neumann 2004).

At any stage of the Market Engineering process, it is decided whether to proceed with the following step or better to repeat the prior one. The use of prototypes is encouraged at any stage of the process. In Figure 1, several design methods are associated with any phase of the process. Those methods do not cover all aspects of the Market Engineering process and for some phases more than one method exist. For instance, the methods of primal/dual programming (Kalagnanam et al. 2003) and mechanism design (Jackson 2002) are substituting each other as they address the same problem in the conceptual design of the market mechanism, while the method of blueprinting complements the latter methods as it aims at the transformation of the conceptual design model into a software model. For a detailed description of how these methods can be sequenced effectively see (Holtmann 2004) and (Neumann 2004).

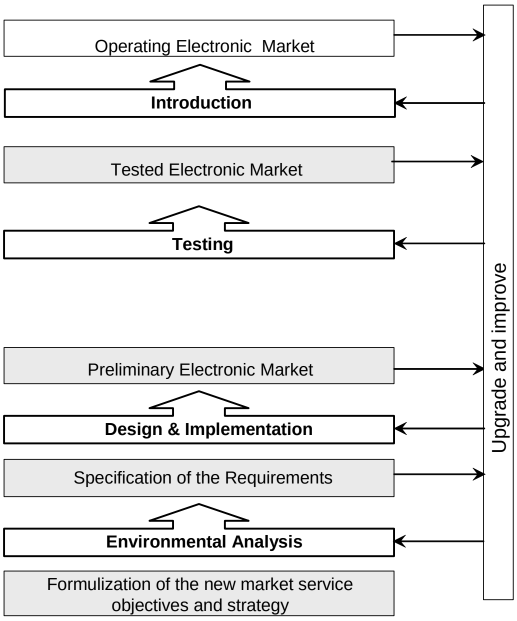

Operating Electronic Market
Introduction
Tested Electronic Market
Testing
Preliminary Electronic Market
Design & Implementation
Specification of the Requirements
Environmental Analysis
Formulization of the new market service objectives and strategy
Upgrade and improve

Figure 1: Design Process and Design Methods

In the following we will use the early stages of the Market Engineering process to tackle the design of a market mechanism for Grids. As aforementioned, any Market Engineering process starts with the objective the market should satisfy. The introduction of this paper hinted at the main objective for Grid markets stating that markets need to set the right incentives such that participants contribute idle resources. In economic terms this coincides with the notion of allocative efficiency.

3. Environmental Analysis

The environmental analysis starts first with a more concise description of the scenario where the Grid market is planned to work (3.1). Based on this, the requirements for this market are deduced (3.2). Related work is evaluated in terms of those requirements, setting up a list a market for this scenario needs to satisfy (3.3).

3.1. Motivating Scenario

The motivating scenario for Grid markets is marked by a video surveillance environment. Imagine an office building which accommodates several organizations, each operating video surveillance cameras. The aim of the surveillance system is to detect persons and track their movement across the facility. The analysis of the resulting video streams requires vast amounts of computing power. However, to avoid each organization having to deploy its own, separate cluster and to increase resource utilization, the organizations deploy one common inter-organizational compute Grid. In order to perform target separation, the analysis of a video stream performs a set of tasks which run sequentially in a pipeline: motion detection, body detection, direction estimation – in which direction is the person moving? – and field of vision detection which prevents direction lines from cameras to cross walls and other barriers and result in false locations of targets.

Each of these tasks can be encapsulated in a Grid service which can then be instantiated multiple times, possibly spread across multiple physical machines. Each video stream runs sequentially through these Grid services as illustrated in figure 1.

3.2. Requirement Analysis

The video surveillance Grid service application poses a number of requirements towards an economic market mechanism (Stoesser et al. 2007c). The most important requirement is a technical one that imposes several stringent economic requirements on the mechanism. In essence, the technical requirement refers to the deployment of applications. Generally, applications can be deployed by directly accessing resources that are distributed over the network or by invocating of a Grid service that encapsulates the respective resources over standardized interfaces. Those two alternative ways of deployment give rise to totally different requirements.

From a technical point of view resource are extremely easy to describe, as there exist a finite set of resources. A resource may be defined by the operating system (e.g. Linux OS), number of CPUs (e.g. 4 * x86 CPU), memory (e.g. 128MB RAM), etc. The GLUE schema provides a standardized vocabulary for describing computing elements. The standardization of resources offers an easy way to semantically describe them. This in turn alleviates resource discovery, as matchmaking is straightforward.

Services on the other hand are extremely difficult to describe, as the service space is infinite. Recall the services of our video surveillance scenario, which are quite unique. In the case of raw services, i.e. resource-near services, standardized languages such as JSDL exist. Nonetheless, the indefinite search space tremendously exacerbates service description and likewise service discovery.

Both resources and services are provided on the basis of a Quality of Service (QoS) description. For resources, the QoS description is very easy, as the agreed properties and the duration of resource access matter. For services QoS is more difficult, as not only time aspects play a role but also precision and accuracy of the services. Precision and accuracy definition depends on the service and cannot generically be standardized. This has also ramifications on monitoring. While monitoring resource access is relatively easy, the monitoring of very complex services becomes in particular demanding when services are intertwined.

The deployment of applications is easy if it is orchestrated via Grid services using existing Grid middleware. When relying on resources, the executables need to be transferred as well. Resources can be deployed as services – in this case the resource providers have to guarantee the completion of the service at a given point of time. Likewise services can be deployed as resources, where the executables need to be transferred to the resource fabric.

From an economic perspective *resource markets* (either deployed as resource or as service) are promising for automation via an organized electronic market. There are standardized items for sale that potentially attracts many buyers and sellers. Services have again a disadvantage as demand is highly specialized, where only few potential buyers and even sellers exist.

In the motivating scenario, complex services are broken down into raw services. From the above discussion this results in the following technical requirements on the market mechanism applicable for Grids.

- **Service Compatibility:** Service compatibility refers to the deployment of the computing element. As aforementioned, deployment of services requires from the provider to guarantee the provision within a certain time period. A market mechanism for Grids, thus, needs to include time attributes.
- **Immediacy:** The timing of the allocation is crucial. The allocation of services in the video surveillance scenario is time critical as the observed objects are dynamic in relatively constant movement and the workflow of services is sensitive to the order of execution [15]. For instance, the direction estimation process may not be executed prior to the body detection process. Consequently, mechanism is needed which allocates requests and offers in very short amount of time.
- **Bundles:** When allocating services, there is a need to allow bids on bundles generated from multiple raw services. For example, a service can be deployed only if two raw services are allocated to it (e.g. a certain program library and computational resources). Receiving only one leg of the bundle is worthless, as the service cannot be executed. Thus, the market mechanism needs to facilitate bids on bundles in order to alleviate the exposure problem.

In addition to those requirements stemming from the technical capabilities of Grid, there are also economic requirements present, a market mechanism for Grid environments need to satisfy:

- **Allocative efficiency:** This requirement addresses that there should be no “waste” of resources; the system is supposed to make optimal use of its resources. To avoid monopolistic power of either market side, there is a need for double-sided exchanges
- **Incentive compatibility:** The market mechanism should be incentive compatible meaning that participants cannot benefit from cheating the mechanism, i.e. reporting any valuation for a service other than their true valuation. Incentive compatible mechanisms are very powerful, as it rules out strategic gaming considerations of the participants.
- **Individual rationality:** Another requirement from classical mechanism design is called individual rationality. A mechanism is said to be individually rational if its users cannot suffer any loss in utility from participating. This requirement is necessary in order to attract potential partners to attract participants.
- **Budget balance:** The payment scheme of a budget-balanced mechanism needs to be designed such that it does not require any subsidies; the payments to service providers are covered by the payments from service requesters. Any mechanism must be both individually rational and budget-balanced in order to be sustainable over time.

Essentially, the first requirement “allocative efficiency” can be perceived as objective function, while the latter are constraints to the problem. In the following, we are interested in market mechanisms that satisfy the technical and economic requirements.

3.3. Related Work

As aforementioned, the idea to employ market mechanisms for distributed computing environments is not new. Relatively new is the idea to use markets for Grids. Among the first, Buyya (Buyya et al. 2001), Wolski et al. (Wolski et al. 2003), and Subramoniam et al. (Subramoniam et al. 2002) motivated the use of auctions and negotiations for Computational

Grid. In their first paper, Wolski et al. (Wolski et al. 2003) suggested the use of traditional auction formats such as English auctions. Eymann et al. (Eymann et al. 2003) introduce a decentralized bargaining system for resource allocation. Regev et al. (Regev et al. 1998) propose the use of a Vickrey auction for allocation computational resources in distributed systems.

Auction	*Immediacy*	*Bundle Bids*	*Service Compatibility*	*Participants*	*Complexity*
(Eymann et al. 2003)	iterative	no	no	1:1	P
(Buyya et al. 2001)	iterative	no	no	1:n	P
(Wolski et al. 2003)	iterative	no	no	1:n	P
(Regev et al. 1998)	one-shot	yes	no	1:n	NP
(Subramoniam et al. 2002)	one-shot	yes	no	1:n	NP
(Wellman et al. 2001).	iterative	yes	yes	1:n	NP
(Conen 2002).	iterative	yes	yes	1:n	NP
(Parkes et al. 2001)	one-shot	yes	no	n:m	NP
(Biswas et al. 2003)	iterative	yes	no	n:m	NP
(Bapna et al. 2006)	one-shot	yes	yes	1:n	NP/P
(Schnizler et al. 2006)	one-shot	yes	yes	n:m	NP

Table 1: Literature overview

Although often suggested is the effectiveness of traditional bargaining and auction systems in Grid environment conceivably delimited, as the trading objects are traded as unbundled standardized commodities. As a consequence, these auction formats fail to express demand on bundles – exposing the buyers and sellers, respectively, to the risk of receiving only a subset of the bundle without the other. To avoid such an exposure risk of the buyers, Subramoniam et al. (Subramoniam et al. 2002) employ the use of ascending bundling auctions. Nonetheless, the trade goods (i.e. resources) are still considered to be *standardized commodities*. Standardization of the resources (or raw services, respectively) would either imply that the number of resources (raw services) are limited compared to the number of all possible ones or that there are extremely many mechanisms, which are likely to suffer under meager participation. Both implications result in rather inefficient allocations.

Reviewing the requirements upon the mechanism (cf. subsection 3.2), it becomes obvious that the previous described mechanisms fail to satisfy these requirements. Most of the mechanisms proposed in literature do not meet the *immediacy* requirement in a way that they are either iterative in nature or NP hard to compute. Both properties rule out a use in highly interactive markets for different reasons. While iterative mechanisms are unsuited due to the fact that they frequent user feedback is needed, NP hard mechanisms consume too much time to be of use in large scale markets. Beside the requirement of *immediacy*, the negligence of *time attributes* for bundles diminish the use of the proposed market mechanisms.

To account for time attributes, Wellman et al. model single-sided auction protocols for the allocation and scheduling of resources under consideration of different time constraints (Wellman et al. 2001). Conen goes one step further by designing a combinatorial bidding procedure for job scheduling including different running, starting, and ending times of jobs on a processing machine (Conen 2002). Furthermore, Chun et al. (Chun et al. 2005) introduce a combinatorial auction for allocating resources in a sensornet. Likewise Bapna et al propose an auction that allows to bid on bundles and on time slots. In addition, the authors also suggest a heuristic that approximates the outcome of the auction in polynomial time. However, these approaches are single-sided and hence do not create competition on both sides.

Demanding competition on both sides suggests the development of a combinatorial exchange. In literature, Parkes et al. introduce the first combinatorial exchange as a single-shot sealed bid auction (Parkes et al. 2001). As payment scheme, Vickrey discounts are approximated. Biswas and Narahari (Biswas et al. 2003) propose an iterative combinatorial exchange based on a primal/dual programming formulation of the allocation problem. By doing so, the preference elicitation problem can be alleviated, as the bidders can restrict their attention to some preferred bundles in contrast to all $2^G - 1$ possible combinations. Obviously, both approaches neither accounts for time nor for immediacy demands are thus not directly applicable for the problem at hand.

The MACE mechanism developed by Schnizler et al. satisfies all requirements but is NP hard and cannot account for Grids with more than 500 participants within a reasonable time span (Schnizler et al. 2006). Accordingly, this mechanism can be used for batch applications, but not for those applications that require *immediacy* as is the case in the motivating scenario.

This paper intends to tailor a mechanism for allocating Grid by converting the aforementioned approaches into a combinatorial exchange that provides immediacy and also incorporates time constraints.

4. Tailoring a Market for Grid Services[1]

Setting-up of a market for raw services requires that the service requesters (i) can specify their demand and (ii) can actually value the services demanded. In principle, both requirements give rise to tricky research questions itself.

For example, there is on-going research on applying AI-techniques for predicting the resource requirements of future services (Ali et al. 2004). The intuition for those techniques is that the resource needs of future services follow the needs of similar services in the past. Similarity can thereby be established in terms of algorithms, data structures and sizes, etc. Concerning the valuation issue, research even lags more behind. Current proposals use service requirements and duration in an opportunity cost approach in order to come up with some quantitative prices. Current research-in-progress work proposes a process for automated bidding that address both problems in their totality (Neumann et al. 2006). Building on this research it is not too farfetched to assume that the resources/raw services needed for the service are known in advance by the service requester. More specifically, a service requester *j* who would like to submit a job to the Grid reports the job's characteristics (v_j, $\underline{c}_j$, $\underline{m}_j$, s_j, e_j) to the market mechanism where $v_j \in \Re^+$ denotes *j*'s maximum willingness to pay per unit of computing power and time slot, $\underline{c}_j \in \aleph$ and $\underline{m}_j \in \aleph$ the minimum required amount of computing power and memory respectively, and $s_j \in \aleph$ and $e_j \in \aleph$ specify the job's estimated runtime. In the following the terms "service requester" and "job" are used as synonym.

[1] Parts of this section are taken from
Stoesser, J., and Neumann, D. "GREEDEX – A Scalable Clearing Mechanism for Utility Computing," Working Paper, 2007a.

We require the market mechanism to make atomic allocations in the sense that each job can only be executed if there are sufficient resources available in all requested time slots. Jobs can potentially be migrated between several nodes over time but each job can only be executed on one node at a time.

A service provider *n* who would like to contribute a node to the Grid reports the node's characteristics (r_n, $\bar{c}_n$, $\bar{m}_n$, ε_n, λ_n) to the market mechanism where $r_n \in \Re^+$ specifies this node's (pretended) reservation price per unit of computing power and time slot, $\bar{c}_n \in \aleph$ and $\bar{m}_n \in \aleph$ the maximum amount of computing power and memory available on this node, and $\varepsilon_n \in \aleph$ and $\lambda_n \in \aleph$ the time frame during which the node can be accessed. Given sufficient resources, we assume that each node is able to virtually execute multiple jobs in parallel, for instance by using virtualization middleware (e.g. using Xen Hypervisor).

Example: Suppose the following resource requests and offers have been submitted to the system:

Table 1: Sample resource requests and offers

Job *j*	v_j	$\underline{c}_j$	$\underline{m}_j$	s_j	e_j	Node *n*	r_n	$\bar{c}_n$	$\bar{m}_n$	ε_n	λ_n
J1	12	54	126	1	7	N1	4	84	71	2	10
J2	4	85	32	3	7	N2	7	100	101	1	9
J3	12	43	43	1	8						
J4	16	35	43	2	7						
J5	11	47	37	3	7						
J6	6	31	19	2	7						

Job J1 requests to be run in time slots 1 to 7 and requires a minimum of 54 units of computing power and 126 units of memory in each time slot. J1 is willing to pay up to $12 per unit of computing power and time slot, that is $12 * 54 * 7 = $4,536 in total. Node N1 offers 84 units of computing power and 71 units of memory in time slots 2 to 10 and requires a reservation price of $4 per unit of computing power and time slot.

4.1. Winner Determination Problem

Let *J* be the set of resource requests, *N* the set of resource offers, and $T := \{t \in \aleph \mid s_j \le t \le e_j, j \in J\} \cup \{t \in \aleph \mid \varepsilon_n \le t \le \lambda_n, n \in N\}$ the set of time slots across all requests and offers, i.e. the allocation problem's time horizon. Then the winner determination problem which solves the allocation problem *exactly* can be formalized as the following integer program:

$$\max_X V := \sum_{j \in J} \underline{c}_j \sum_{t \in T} \sum_{n \in N} x_{jnt} \left(v_j - r_n\right)$$

$$s.t. \quad x_{jnt} \in \{0,1\}, s_j \le t \le e_j, \varepsilon_n \le t \le \lambda_n, v_j \ge r_n, j \in J, n \in N \qquad (C1)$$

$$\sum_{n \in N} x_{jnt} \le 1, s_j \le t \le e_j, j \in J \qquad (C2)$$

$$\sum_{j \in J} x_{jnt} \underline{c}_j \le \bar{c}_n, \varepsilon_n \le t \le \lambda_n, n \in N \qquad (C3)$$

$$\sum_{j \in J} x_{jnt} \underline{m}_j \le \bar{m}_n, \varepsilon_n \le t \le \lambda_n, n \in N \qquad (C4)$$

$$\sum_{u=s_j}^{e_j} \sum_{n \in N} x_{jnu} = \left(e_j - s_j + 1\right) \sum_{n \in N} x_{jnt}, s_j \le t \le e_j, j \in J \qquad (C5)$$

The objective of this integer program is to maximize welfare *V*, the total difference between the requesters' valuations and the providers' reservation prices across all time slots. Constraint (C1) introduces the binary decision variable *x* and ensures that a job can only be allocated to a node which is accessible during the right time slots and whose reservation price does not exceed the job's willingness to pay. Furthermore, a job can only be allocated to at most one node at a time (C2). Constraints (C3) and (C4) specify that the jobs allocated to one node at a time are not allowed to consume more resources than are available on this node. Constraint (C5) enforces atomicity, i.e. a job is either fully executed or it is not executed at all.

Example: Exactly allocating the sample requests and offers above generates the following schedule:

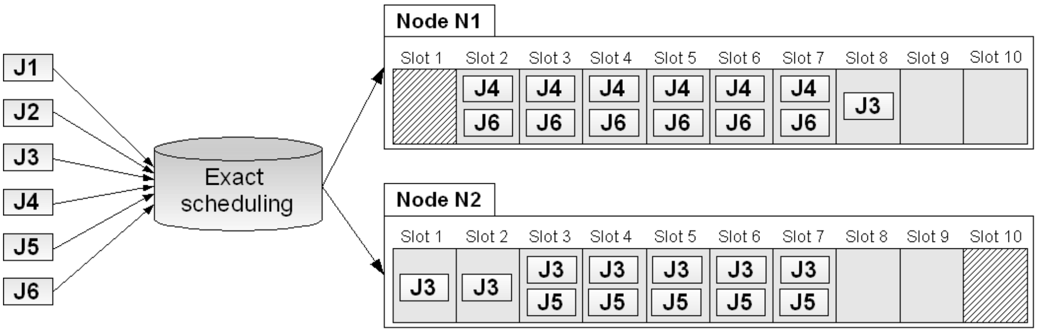


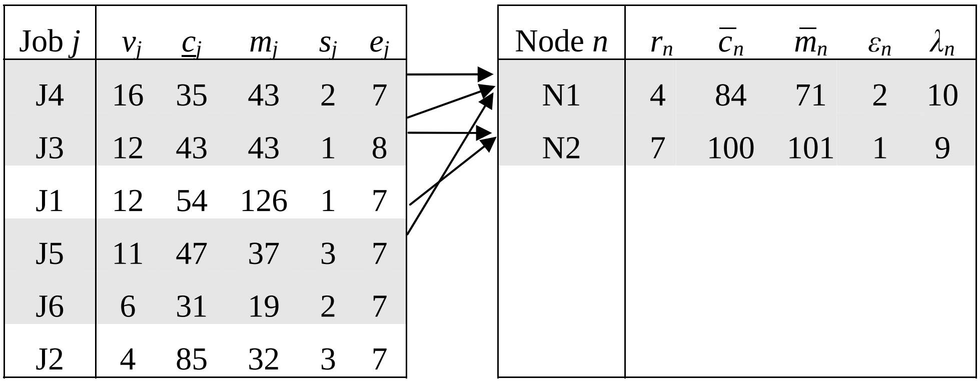
Figure 1: Sample allocation schedule

The allocation problem at hand is an instance of a multi-dimensional knapsack problem and as such clearly NP-hard. Heuristics have the desirable property of generating suboptimal allocations fast. We define the following greedy heuristic:

1. Sort jobs $j \in J$ in non-ascending order of their reported willingness to pay v_j. Sort nodes $n \in N$ in non-descending order of their reported reservation prices r_n. If there are ties between jobs (nodes) when generating the ranking, the job (node) which has been reported first is preferred.
2. Starting with job j with the highest ranking (i.e. the highest reported willingness to pay), allocate j to the nodes $n_1, \ldots, n_k$ with the highest ranking (i.e. the lowest reported reservation prices) which can together accommodate j.
3. Repeat the allocation procedure of step 2 with the next job in the ranking until there are no more jobs which can be allocated to the available nodes.

This heuristic truly implements a greedy allocation scheme: it tries to greedily maximize the term $v_j - r_n$ in the objective function of the exact allocation problem above.

Example: For the sample requests and offers at hand, the greedy heuristic is illustrated in Figure 2:



Job j	v_j	$\underline{c}_j$	m_j	s_j	e_j
J4	16	35	43	2	7
J3	12	43	43	1	8
J1	12	54	126	1	7
J5	11	47	37	3	7
J6	6	31	19	2	7
J2	4	85	32	3	7

Node n	r_n	$\overline{c}_n$	$\overline{m}_n$	ε_n	λ_n
N1	4	84	71	2	10
N2	7	100	101	1	9

Figure 2: The greedy heuristic applied to the sample requests and offers

The winning jobs and nodes are highlighted. Job J4 can be allocated to node N1 since N1 offers sufficient resources over all required time slots and its reported reservation price is less than J1's reported willingness to pay. In time slot 1, J3 can be allocated to node N1. However, in time slots 2 to 8, there is not sufficient residual capacity left due to the execution of J4. So J3 is subsequently allocated to the next available node N2. J1 cannot be executed at all due to its excessive memory requirements. The heuristic proceeds until the ranked list of jobs ends and finally happens to generate the same allocation schedule as the exact mechanism for the example at hand.

4.2. Pricing Schemes

The allocation algorithm of market mechanism intends to achieve some global or social aim, in this case maximize social welfare. In achieving this goal, it depends on the resource requesters and providers to report their true valuations and resource characteristics. These participants, however, are assumed to be rational and self-interested agents trying to maximize their individual benefit which may not per se be aligned with the social aim. Pricing schemes are introduced and closely linked to the allocation algorithm so as to induce participants to indeed report *truthfully* about their valuations and characteristics while maintaining other economic criteria (Stößer et al. 2007, Schnizler et al forthcoming). In this section, we will introduce a truthful pricing scheme on the requester's side and a proportional pricing scheme on the sellers' side.

Critical-value-based Pricing

The rationale behind the pricing of resource requests is similar to the well-known Vickrey-Principle: the payment of each winning job amounts to the lowest (total) willingness to pay it would have needed to report in order to still remain in the allocation, keeping all other resource requests and offers fixed. This is also called the critical value ϕ_j of job j (Lehmann et al. 2002, Mu'alem and Nisan 2002). This is illustrated best by looking at the example at hand.

Example: The critical values and resulting payments for the winning jobs J3, J4, J5 and J6 are given in Table 2:

Table 2: Critical-value-based pricing of the sample resource requests

Job j	J3	J4	J5	J6
v_j	12	16	11	6
ϕ_j	7	7	7	4
$p_{greedy,j}$	2,408	1,470	1,645	744

Job J3 would have needed to report a willingness to pay of at least \$7 – the reservation price of node N2 – in order to still remain in the allocation schedule. Consequently, ϕ_j = \$7 and the resulting greedy price of J3 is $p_{greedy,j} = (e_j - s_j + 1)\underline{c_j}\phi_j$ = 8 * 43 * \$7 = \$2,408. The prices of J4, J5 and J6 are determined accordingly. Jobs J1 and J2 have not been allocated and consequently no payment is required. Overall, the pricing scheme generates revenue of \$6,267.

Truthfulness with respect to a job's resource requirements is straightforward. If a job's resource requirements are understated, the job will not be finished; it has to pay for the used resources but these are of no value. Overstating a job's requirements either increases the job's payment or the job is not scheduled at all. Regarding the reported willingness to pay, the main feature of the proposed pricing scheme is its ability to generate truthful prices of resource requests: It is a (weakly) dominant strategy for resource requesters to report their true valuations. This is a strong and desirable feature as it tremendously simplifies the resource requesters' strategy space. For the individual participant, there is no need to reason about the other participants' strategies. Instead, she can "simply" report her true valuation. Note that we assume participants to know their true valuations. Of course this yields complexity on a different level, being preference elicitation. However, this is outside the scope of this paper.

The critical value of a job essentially hinges on the competition of other jobs for the same resources. If there is no competition, the job is only required to pay the reservation price. Otherwise, the job at least needs to outbid these competing jobs. Consequently, critical-value-based pricing implements a desirable dynamicity from the providers' point of view. Providers do not need to constantly monitor the demand-side of the market to set appropriate prices. Instead, they can simply delegate this to the market mechanism. If there is sufficient competition such that critical values are above reservation prices, critical-value-based pricing of resource requests potentially generates a surplus, i.e. overall payments exceeding the Grid providers' reservation prices.

Proportional Pricing

Like the service requesters, these providers are assumed to be selfish agents trying to report to the mechanism so as to maximize their individual benefit. Grid providers clearly do neither have an incentive to understate nor to overstate their availability of resources; understating may reduce the generated revenue while overstating can be monitored and punished. Unfortunately, a truthful pricing scheme such as the critical-value-based pricing (Stoesser et al. 2007b) is not applicable to service offers; it requires a binary decision in the sense that either all services of an offer are allocated or none at all. However, in our model, we allow the partial use of nodes. In order to at least approximate truthful payments, we thus fall back to payments proportional to each provider's contribution of processing power to the allocation schedule. Let

$$S := \sum_{j\in J} p_{greedy,j} - \sum_{j\in J} \sum_{n\in N} \sum_{t\in T} x_{jnt} \frac{\underline{c}_j}{\overline{c}_n} r_n$$

be this surplus. Then provider *n* will receive a payment amounting to

$$p_{greedy,n} = S \cdot \frac{\sum_{j\in J} \sum_{t\in T} x_{jnt} \underline{c}_j}{\sum_{v\in N} \sum_{j\in J} \sum_{t\in T} x_{jvt} \underline{c}_j},$$

i.e. *n* will receive a share of the surplus according to its share in the total contribution of processing power across all providers.

Example: In the example at hand, nodes N1 and N2 have reservation prices of $1,756 and $3,752 respectively for the allocation schedule. Thus there is a surplus of $6,267 – $1,756 – $3,752 = $759. In total, 975 units of processing power are provided, 439 by N1 and 536 by N2. Consequently, N1 receives a total payment of $1,756 + 0.45 * $759 = $2,097.55 and N2 receives $3,752 + 0.55 * $759 = $4,169.45.

As stated above, there are further economic design criteria besides truthfulness. The two most prominent and desirable are *budget-balance* and *individual rationality*.

A mechanism is said to be budget-balanced if its pricing scheme does not need to be subsidized by outside payments, i.e. the payments from the service providers cover the payments made to Grid providers. The pricing scheme of the proposed mechanism is specifically designed so as to obtain strongly budget-balanced payments. Any surplus from the demand-side is distributed to the supply-side of the market. After settling the market, there are neither payments left, nor does the market need to be subsidized with outside payments.

Individual rationality is satisfied if no participant can suffer a loss from participating in the mechanism. The proposed mechanism is individually rational, as no resource requester needs to pay anything in case her request is not accepted, nor does she have to pay more than her reported willingness to pay in case her request is served. Each Grid provider is guaranteed to receive at least her reported reservation price for her contributed resources. Budget-balance and individual rationality are hard feasibility constraints and must be satisfied in order for the market to be sustainable.

5. Concluding Remarks

In this paper we design a market mechanism for Grids according to the Market Engineering process. Market Engineering denotes the systematic and theoretically founded design of markets. By relying on the prescription provided by the Market Engineering process, it is assumed that the design process may result in better markets tailored to the needs of the potential participants. To design a market for computer resources, the Market Engineering process was adopted to create adequate market mechanisms.

The requirement analysis derives a set of requirements on a market mechanism, consisting of a bidding language, a winner determination and a pricing scheme. When talking about Grids, it is often referred to services which can be either very complex or close to the resources. As important requirements for trading resource-near raw services addresses the timing of the allocation process. In essence, services often require an immediate allocation. As the analysis of related work shows, none of the existing mechanisms can meet all requirements for trading raw services.

In this paper, we define a market mechanism that is tailored to the needs of Grid trading. The mechanism consists of a bidding language which allows the definition of bundles. The winner determination problem is NP hard, thus we propose a greedy heuristic which solves the problem in P. The pricing scheme is divided into two parts. While the prices on the requester’s side are strategy proof being based on critical value pricing, the prices on the provider’s side are not. This stems from the impossibility to set up a strategy proof exchange that is budget balanced. In this paper we suggest a proportional pricing scheme which distributes the prices gained from the requesters among the provider’s proportionally. Further evaluation of the mechanism concerning the economic properties and field studies which demonstrate the applicability of the mechanisms are needed in the future.

References

Bapna, R., Das, S., Garfinkel, R., and Stallaert, J. "A market design for grid computing," *INFORMS Journal of Computing*) 2006, p fortcoming.

Biswas, S., and Narahari, Y. "Iterative dutch combinatorial auctions," *Annals of Mathematics and Artificial Intelligence*) 2003, p forthcoming.

Buyya, R., Stockinger, H., Ghiddy, J., and Abramson, D. "Economic models for management of resources in peer-to-peer and grid computing," International Conference on Commercial Applications for High Performance Computing, Denver, CO, 2001.

Chun, B.N., Buonadonna, P., AuYoung, A., Ng, C., Parkes, D.C., Shneidman, J., Snoeren, A.C., and Vahdat, A. "Mirage: A Microeconomic Resource Allocation System for SensorNet Testbeds," Workshop on Embedded Networked Sensors, 2005.

Conen, W. "Economically coordinated job shop scheduling and decision point bidding - an example for economic coordination in manufacturing and logistics," Workshop on Planen, Scheduling und Konfigurieren, Entwerfen, Freiburg, 2002.

Eymann, T., Reinicke, M., Ardaiz, O., Artigas, P., Díaz de Cerio, L., Freitag, F., Messeguer, R., Navarro, L., and Royo, D. "Exploring Decentralized Resource Allocation in Application Layer Networks," Agent Based Simulation, 2003.

Grant, D. "Design Methodology and Design Methods," *Design Methods and Theories* (13:1) 1979, pp 46-47.

Holtmann, C. "Organisation von Märkten - Market Engineering für den Wertpapierhandel," in: *Economics and Business Engineering*, University of Karlsruhe (TH), Karlsruhe, Germany, 2004.

Jackson, M.O. "Mechanism Theory," in: *Encyclopedia of Life Support Systems*, UNESCO - online, 2002.

Kalagnanam, J., and Parkes, D.C. "Auctions, Bidding and Exchange Design," in: *Supply Chain Analysis in the eBusiness Era*, S.D.W.a.Z.M.S. David Simchi-Levi (ed.), Kluwer Academic Publishing, 2003, p. forthcoming.

Neumann, D. "Market Engineering - A Structured Design Process for Electronic Markets," in: *Fakultät für Wirtschaftswissenschaften*, Universität Karlsruhe (TH), Karlsruhe, 2004.

Neumann, D., Lamparter, S., and Schnizler, B. "Automated Bidding for Trading Grid Services," Proceedings of the European Conference on Information Systems (ECIS), 2006.

Pahl, G., and Beitz, W. *Engineering Design* The Pitman Press, Bath, UK, 1984.

Parkes, D.C., Kalagnanam, J., and Eso, M. "Achieving budget-balance with vickrey-based payment schemes in exchanges," International Joint Conference on Artificial Intelligence, 2001, p. 1161–1168.

Regev, O., and Nisan, N. "The POPCORN market - an online market for computational resources," First international conference on Information and computation economies, ACM Press, Charleston, South Carolina, 1998, pp. 148 - 157.

Schnizler, B., Neumann, D., Veit, D., and Weinhardt, C. "Trading Grid Services - A Multiattribute Combinatorial Approach," *European Journal of Operation Research*) 2006, p forthcoming.

Stoesser, J., and Neumann, D. "GREEDEX – A Scalable Clearing Mechanism for Utility Computing," Working Paper, 2007a.

Stoesser, J., Neumann, D., and Anandasivam, A. "A Truthful Heuristic for Efficient Scheduling in Network-Centric Grid OS," European Conference on Information Systems (ECIS 2007), St. Gallen, 2007b, p. forthcoming.

Stoesser, J., Roessle, C., and Neumann, D. "Decentralized Online Ressource Allocation for Dynamic Web Service Applications," IEEE Joint Conference on E-Commerce Technology (CEC'07) and Enterprise Computing, E-Commerce and E-Services (EEE '07), Tokyo, Japan, 2007c.

Subramoniam, K., Maheswaran, M., and Toulouse, M. "Towards a micro-economic model for resource allocation in grid computing systems," IEEE Canadian Conference on Electrical & Computer Engineering, 2002.

Sutherland, I.E. "A futures market in computer time," *Communications of the ACM* (11:6) 1968, pp 449-451.

Weinhardt, C., Holtmann, C., and Neumann, D. "Market Engineering," *Wirtschaftsinformatik* (45:6) 2003, pp 635-640.

Wellman, M.P., Walsh, W.E., Wurman, P., and MacKie-Mason, J. "Auction protocols for decentralized scheduling," *Games and Economic Behavior* (35:271-303) 2001.

Wolski, R., Brevik, J., Plank, J., and Bryan, T. "Grid resource allocation and control using computational economies," in: *Grid Computing - Making The Global Infrastructure a Reality*, John Wiley & Sons, 2003, p. chapter 32.