Association for Information Systems
AIS Electronic Library (AISeL)

Wirtschaftsinformatik Proceedings 2005 | Wirtschaftsinformatik

February 2005

SettleBot: A Negotiation Model for the Agent Based Commercial Grid

Florian Lang
University of Erlangen-Nuremberg

Follow this and additional works at: http://aisel.aisnet.org/wi2005

Recommended Citation
Lang, Florian, "SettleBot: A Negotiation Model for the Agent Based Commercial Grid" (2005). *Wirtschaftsinformatik Proceedings 2005*. 9.
http://aisel.aisnet.org/wi2005/9

This material is brought to you by the Wirtschaftsinformatik at AIS Electronic Library (AISeL). It has been accepted for inclusion in Wirtschaftsinformatik Proceedings 2005 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

In: Ferstl, Otto K, u.a. (Hg) 2005. *Wirtschaftsinformatik 2005: eEconomy, eGovernment, eSociety*; 7. Internationale Tagung Wirtschaftsinformatik 2005. Heidelberg: Physica-Verlag

ISBN: 3-7908-1574-8

© Physica-Verlag Heidelberg 2005

SettleBot: A Negotiation Model for the Agent Based Commercial Grid

Florian Lang
University of Erlangen-Nuremberg

Abstract: Market-driven sharing of distributed computational resources requires coordination support that can be provided by distributed problem solving (software agent technology). Multiple-issue negotiation among autonomous software agents allows the efficient alignment of resource consumers' demand profiles and the service capabilities of resource providers. To address the inefficiencies of negotiations on imperfect markets, the negotiation model suggested by the SettleBot research effort includes both self-interested negotiations driven by a heuristic strategy and a joint-gains approach to win/win-negotiations. While finding joint gains under imperfect information is a well-known problem with approaches relating to simulated annealing as common approximate solutions, self-interested negotiations in a dynamically evolving environment require intelligent agents that retrieve, process and leverage knowledge about the world state. Superior strategy solutions in given market scenarios are identified using a genetic learning algorithm.

Keywords: Grid Computing, Negotiation, Multi-Agent Systems

1 Commercial Grids

Successful grid implementations in high performance scientific applications have led to grid technology as a promising approach to cost cutting in professional business environments. Business cases around the world show that grid computing has already found its way to the companies' data centers. The goal of grid enabled business applications is to reduce the cost and boost the efficiency of heavy duty application services that rely on highly available network infrastructure, data processing and storage. The principle of sharing computational resources dynamically among applications is enabled by a grid infrastructure (grid fabric) and a grid management appliance that allocates idle resources to service consuming applications.

While cooperative sharing of resources can be readily assumed in scientific areas of application, it cannot provide for an allocation approach in competitive environments. Self-interested players will only share their resources across profit-

centers or companies, if they are compensated for the exchanged services. The SettleBot project aims at providing a market based allocation mechanism for the exchange of grid services in competitive environments (referred to as "grid market" in the remainder of this article). A commercial grid allows the virtualization of resources across separate budget entities (profit centers, companies). Idle resources become chargeable grid service offers. On the client side, excess demand for service resources may be satisfied by finding and using idle resources via the commercial grid. Thus, a subsystem of the grid may efficiently size its resource pool well below peak load requirements.

A grid market must support flexible pricing to ensure an efficient match of supply and demand. Beginning with the Harvard PDP-1 auction in 1968 (see [Suth68]) many research efforts have evaluated auctioning as an economic mechanism for the valuation and exchange of computational resources (see for example [Chen+02; Gagl+95; ReNi98; Wols+01]). However, auction mechanisms require fixed goods. An auctioneer will offer a fixed service level agreement (service type, execution time, resource usage) that is valued by a bidder's decision function. The SettleBot project extends this approach by providing a mechanism that allows agreeing upon any attribute of the SLA, including price. A grid market that supports such multi-issue negotiation allows efficient resource usage by matching suppliers' and demanders' preference structures with respect to all aspects of a service level agreement. The project's goal is to design negotiating software agents that autonomously negotiate for multiple-attribute grid service contracts.

A computational grid is an IT-infrastructure that integrates a pool of heterogeneous resources and presents them to users as a single machine (virtualization). Besides joining resources that are heterogeneous in terms of platform and performance, grid computing extends cluster computing by considering resource failure a standard operating condition. The grid is a highly dynamic computing infrastructure, where resources are merged and unmerged depending on whether they are available (up and idle) or unavailable (down or busy). While first generation computational grids involved proprietary solutions, second generation grids introduced middleware as a framework to cope with scalability, heterogeneity and efficient resource allocation. Third generation computational grids follow a service-oriented approach, are metadata-enabled like web services and may exhibit autonomic features [Dero+03].

State of the Art-Grid Middleware applies centralized job scheduling algorithms for resource allocation. Job scheduling is a suitable approach for managing resources in control by a central grid manager. Separate budget scenarios where grid resources are distributed among independent resource owners require mutual consent on which resources may be scheduled to run a job and when. In a commercial grid, this mutual consent is reached by a market mechanism implemented in the middleware layer that allows resource valuation and exchange. We consider the commercial grid a third generation computational grid that applies market based resource allocation (see Figure 1).

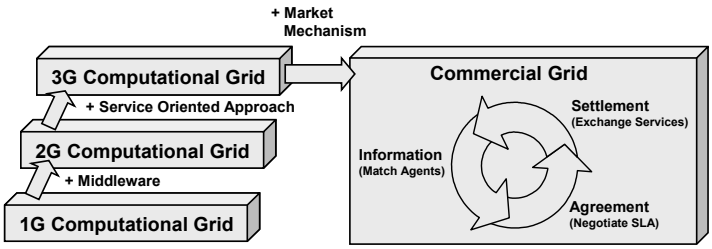


Figure 1: Grid Evolution

Commercial grids allow service consumers to noticeably reduce their application total cost of ownership by outsourcing peak loads to the most cost-efficient source. Given that otherwise idle resources are available for grid service insourcing, cost cutting effects can be dramatic. A resource owner who has got non-storable, idle resources to spare will accept any compensation exceeding transaction cost.

2 Automated Negotiation and Knowledge Processing

The operative interdependencies that arise from interweaving businesses pose major coordination challenges. In competitive environments, coordination as an instrument of managing interdependencies is restricted by contradictory goals of the interdependent parties. Negotiation is an instrument of resolving these conflicts.

Machine negotiation is applied to allow short-termed, flexible contracting among autonomous systems at very low cost per transaction. The SettleBot research effort aims at developing a heuristic negotiation model that deals with the inefficiencies prevailing in real-world settings. SettleBot's agents try to maximize their payoff by applying a heuristic strategy that processes the agents' knowledge about their environment, their opponents and their own goals. Since information about other agents' preferences is limited in competitive settings, automated negotiations driven by heuristic strategies usually result in Pareto-dominated contracts. This downside to heuristic negotiations is met by SettleBot's two-phased negotiation protocol.

In the first phase, the agents apply self-interested negotiation strategies to reach a preliminary agreement. This agreement reflects the agents' individual success in gaining maximum utility with respect to their individual preference models. The first phase may be characterized as a win/lose-negotiation where an agent maximizes its payoff on another agent's expense ("dividing the pie"). The second phase aims at correcting inefficiencies of the preliminary agreement by searching

for joint gains ("enlarging the pie") while preserving the payoff distribution the agents gained from the first phase. Thus, the agent that has a stronger negotiation stance (e.g. little competition by identical or substitutional service offerings), uses a better strategy or leverages other competitive edges like additional or more accurate knowledge, better knowledge processing skills and so forth still "wins" the negotiation while both sides yield additional payoff.

Centralized auctioning mechanisms assume goods or services of equivalent quality that are distinguishable only by price. On commodity markets, the quality of variable goods and services (i.e. agricultural products, bandwidth) is standardized by a common quality statement (commoditization), thereby avoiding the transaction cost of consumers and providers coming to individual terms on variable quality features. Contract variables other than price are non-negotiable on commodity markets, thus obscuring additional profit opportunities ("leaving money on the table"). Multiple-issue negotiation allows preference-driven bilateral contract customization at the attribute level, so contractors are awarded a better fit of consumer needs and provider productivity.

Concerning a resource management system's user interface, user attention must be considered the scarcest resource to be managed. Therefore, hiding the complexities of real-time multi-attribute resource allocation from users is a major design goal. Enabling agents with autonomous knowledge retrieval and processing abilities is a design principle.

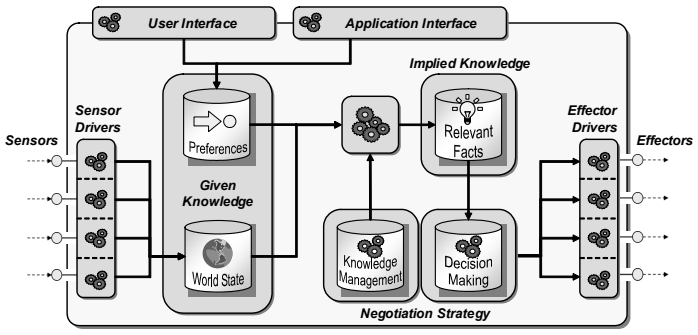


Figure 2: Knowledge Processing Architecture

Transaction support in the SettleBot system aims at reducing user involvement in real-time grid service allocation settings to a budget decision and a non-recurring elicitation of the user's trade-off weight concerning time vs. money. Goals and preferences other than budget and the trade-off weight are extracted by an application interface (see Figure 2). A sensors stack allows an agent to observe dynamically changing environmental conditions that must be considered by the agent's negotiation strategy (competing offers, time, price history, …). Sensor drivers process, aggregate and store these observations.

The world state is the agent's perception of its fluctuating environment. Both preferences and knowledge about the world state form input for the agents negotiation strategy and are forwarded to knowledge management functions that map an agent's strategically relevant knowledge on knowledge bits ("relevant facts", see Figure 2) directly exploitable by the agent's heuristic strategy. The agent's knowledge based decision making includes provider selection, budget constraint modification, fixing reservation values, determining aggressiveness (reluctance to make concessions) and other decisions that may help to increase an agent's payoff when properly made. Effectors turn decisions into actions. The effectors stack determines the agent's behavioral degrees of freedom. An agent uses its effectors to place offers, conclude deals or terminate negotiations.

3 Agent Based Grid Service Transactions

A service level agreement (SLA) allows the definition of mutual guarantees among service consumers and service providers. It covers job variables detailing type and composition of the service to be delivered (service level objectives, exclusions), and deal variables that specify the contracting parties, price, time constraints, penalties and other administrative minutiae.

The structure of an SLA for grid resource allocation depends on the application scenario it is designed for. There is no "one fits all"-solution to SLA design for the commercial grid. However, there are some basic features that are common or at least widely reusable (see Table 1). Besides identifying the contracting parties and the service class, an SLA must enable mutual agreement on the service bundle to be delivered, the time frame in which delivery is due and monetary compensation.

Non-Negotiables	Negotiables
`provider_id`	`packets`
`consumer_id`	`deadline`
`service_id`	`price`

Table 1: SLA Structure

A typical resource bundle needed to provide application services includes computation, storage, traffic and memory. These resources form what is called a "linked value group" [Keny02]. For example, computation without bandwidth is useless, and so is bandwidth without computation. In a scenario where a huge workload of similar jobs (capacity computing), or a single large, finitely divisible job (capability computing) is assigned to distributed resources, each subjob ("packet") con-

sumes a certain resource bundle, type and composition of which depends on the respective application or a specific job.

A central application of commercial resource allocation in competitive environments will be the instant delivery of grid services on-demand, relieving application servers of peak loads in real-time. Therefore, grid service contracts are executable on the spot on a real-time market. A deadline indicates when the subjob must be completed. The SLA structure depicted in Table 1 serves as a contracting framework for resource allocation driven by agent-based "over the counter" negotiations. "Non-negotiables" identify the contractors committed to the SLA and the service class to be delivered. Besides `deadline` and `price` the negotiables section includes the number of `packets` to be delivered. A packet is the smallest, non-divisible fraction of a finitely divisible grid job.

A business transaction consists of an information, an agreement and a settlement phase [Zbor96] (see Figure 3). An electronic market may support one or all of these phases, depending on user requirements, characteristics of the transaction object, transaction cost and availability respectively profitability of support tools. A rough analysis of the market scenario given by a commercial grid implies that decision support by autonomous systems is required in all phases:

Resource failure as well as spontaneous entry and unpredictable exit of service providers lead to a highly dynamic topology of the grid and thus to low market transparency in the information phase. This restricted transparency contradicts the need for "co-allocation", i.e. the simultaneous sourcing of complex application services from multiple providers as required in a commercial grid (see [Chen+02; Czaj+99]). The complexity must be hidden from users in all transaction phases.

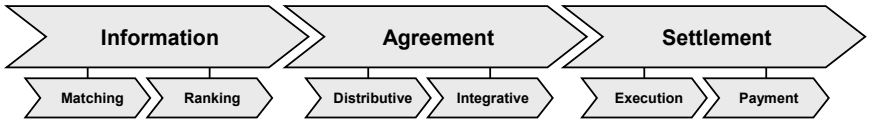


Figure 3: Transaction Phases and Support Functions

Matching consumers and providers in the information phase as well as negotiation driven by preference models involves detailed technical descriptions of supply and demand profiles. Generating these profiles requires expensive expert knowledge, unless they are generated automatically. Moreover, spontaneously evolving needs, e.g. in application server load balancing scenarios, call for near real-time demand communication.

We suggest an agent based approach to transaction support for the commercial grid. A grid market as designed in the SettleBot project consists of three node types representing consumers, providers and intermediaries. Each node type (see Figure 4) hosts an agent that implements the decision support functionality required by the node type. A grid carrier agent (GCA) implements the intermediary

functions that allow the registration of grid services offered by provider agents. A grid service resource includes a grid service factory that instantiates services upon request (see [Fost+02]) and actual physical resources like storage, bandwidth, computation and memory.

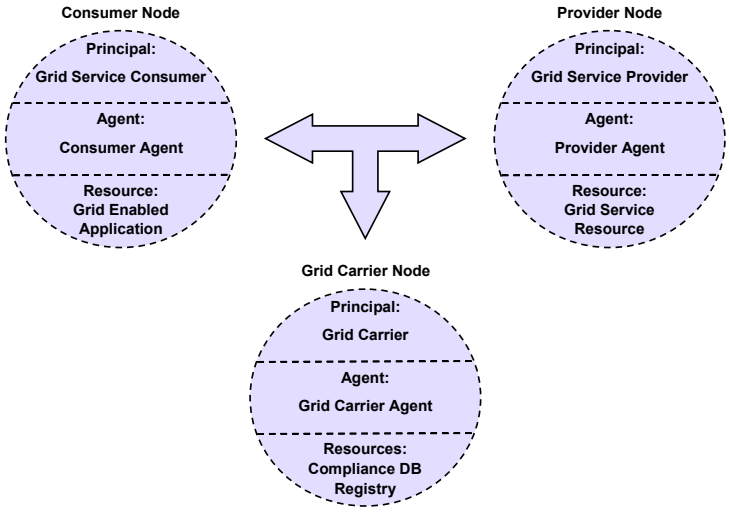


Figure 4: Principals, Agents and Resources of the Commercial Grid

To find offers matching their preference profile, consumer agents retrieve entries from the registry by querying the GCA. A preference driven ranking of matching offers completes the information phase (see Figure 3). In the agreement phase, consumer and provider agents self-interestedly negotiate for distributive agreements that are additionally improved by an integrative search for joint gains. Due to the limited capacity of a single provider, the ability to assign excess demand for finitely divisible application services to multiple external providers improves allocation efficiency. Therefore, a single job may require many transactions and the foregoing negotiations for a set of interdependent contracts. Since grid jobs are considered divisible, they can be decomposed to single packets, setting the maximum number of providers simultaneously involved to the number of packets. The settlement phase of a transaction includes service execution and payment (execution of the SLA).

Peak load management as a grid application leads to instantaneous, dynamic needs for grid services that must be served on short notice. Therefore, thorough decision support by autonomous systems is not a feature but a must in a commercial grid. The following sections explain some of the technical details on grid service transactions supported by the suggested agent based framework.

4 Preferences

Providers and consumers in a commercial grid are assumed to have preferences of one contract alternative over another. Considering the variable attributes of the SLA, a rational provider prefers low resource spending, high revenue and long deadlines, while a rational consumer prefers high resource usage, low cost and short-term service delivery. These preferences are modeled by an ordinal utility function that guides an agent's decisions in the information and the agreement phase.

For n attributes, let $a = (a_1, a_2, ..., a_n)$ and $b = (b_1, b_2, ..., b_n)$ be alternative outcomes with a preferred to b by the principal ($a \succ b$), the agent's utility function u must be designed such that $u(a) > u(b)$. The type of the utility function (multilinear, multiplicative, additive) depends on the degree of dependence among the attributes. An attribute X_1 is considered preferentially independent of an attribute X_2, if the preference $(x_1, x_2) \succ (x_1', x_2)$ is independent of the specific value of x_2. Krantz et al. show that for mutually, preferentially independent attributes, a utility function takes the form $u = f(u_1, u_2, ..., u_n)$ [Kran+71]. For ease of valuation and utility function design, it is desirable to model the utility function in its well understood additive form

$$u = \sum_i w_i \cdot u_i(x_i) \tag{1}$$

An additive utility function requires all attributes to be mutually, preferentially independent [KeRa93]. Mutual preferential independence (MPI) is given if any subset of attributes is preferentially independent of its complement. Among the attributes **`packets`**, **`deadline`** and **`price`**, there is no subset that must be considered preferentially dependent of its complement. The MPI assumption reduces utility function design to determining the subutility functions $u_i(x_i)$ and the weights w_i for each attribute $i = (1, 2, ..., n)$. Since money yields diminishing marginal utility the utility of price can be modeled using a Bernoulli-concave (logarithmic) subutility function. A rational individual's claim for risk compensation (risk aversion) also implies concave subutility functions for the attributes **`packets`** and **`deadline`**.

For generating its subutility functions, an agent needs to derive aspiration values (a_i) and reservation values (r_i) for each of the negotiable attributes from user constraints while also considering the total job size. Since a divisible job must be simultaneously assigned to several grid service providers for efficiency purposes, r_{price} and $r_{packets}$ must be dynamically set during negotiations by breaking down the total values of budget and job size. Since the execution of fractions of the total job (joblets) by more than one provider is not sequential, $r_{deadline}$ in a negotiation equals the total user deadline.

While the reservation values are derived from the user constraints, an agent autonomously sets its aspiration values (goals). The goals of a consumer agent concerning the number of packets it assigns to a provider agent ($a_{packets}$) and the deadline it tries to arrange ($a_{deadline}$) is to maximize the provider agent's load, considering its performance indicator p [packets/sec] while trying to minimize `price` ($a_{price} = 0$).

To value the offers it receives from a provider agent, a consumer agent needs to weight the three attributes' subutilities. A user's internal utility measurement in a grid scenario considers a time versus money trade-off that can be elicited by common methods such as the simple multi-attribute rating technique (SMART, see [WiEd86]) and analytical hierarchical processing (AHP, see [Saat92]).

A grid service provider is a resource owner that frees idle resources, which then become grid service resources. "Freeing" resources means defining preferences that govern the assignment of these resources to the provider agent. The provider agent registers service offers with the grid carrier agent whenever the resources become available. Resource availability is determined by preference rules that allocate resources either automatically, e.g. when the resource utilization drops below a pre-defined threshold, or by asking the grid service provider for feedback on what resources may be freed for what time. In typical peak load outsourcing scenarios, server load changes due to events out of control of the resource owner. To maintain usability and efficiency, unpredictable peak loads and idle times require a set of preference rules that enable provider agents to autonomously decide their actions (e.g. "IF CPU usage has been below 30% for 30 minutes, THEN assign 50% of CPU for the next 10 minutes."). Based on the resource availability prospect given by such rules and the resource consumption of a single service packet, an agent calculates its performance indicator p [packets/sec] and its maximum contract duration $r_{deadline}$ that limits the timeframe of service provision commitments made by a provider agent.

A provider agent dynamically recalculates its $r_{packets}$ during a negotiation, depending on its p [packets/sec] and the deadline set by the currently exchanged offer. r_{price} represents a user's minimum compensation claim for providing grid services. Since a rational provider agent wants to sell minimum packets at maximum prices and long deadlines, its aspiration values are set according to these goals.

5 Transaction Support

5.1 Information Phase

In the information phase, provider agents exchange advertisements with consumer agents via the grid carrier agent's registry that closely relates to a UDDI registry when dealing with web services. Referring to the Globus Toolkit, any grid service publishes service data that allows both to become a stateful service (as opposed to web services) and to publish attributes describing the provider's capabilities [Czaj+03]. When a provider registers with the grid carrier agent's registry, it posts the service data that is relevant for provider selection by consumer agents. Besides data such as a unique identifier by which the provider's grid service factory is addressed (factory handle) and the factory's service deadline $r_{deadline}$, an ad contains the service ID of the service the factory is able to instantiate. Each service ID is bound to a Grid Service Reference (GSR) describing the grid service's interface. Based on the resource availability prospect given by the preference rules and the resource consumption of a service packet, a provider agent additionally calculates its performance indicator p [packets/sec]. The performance indicator is the key selection criterion for consumer agents.

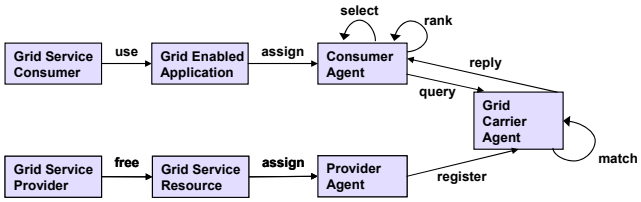


Figure 5: Object Calls in the Information Phase

A consumer agent in search of a specific service queries the GCA's registry for the corresponding service ID and obtains a list of factory handles matching the query along with the performance indicators and service deadlines (see Figure 5). After ranking the provider list by performance indicators, a consumer agent selects the providers to negotiate with from the list. In the example given by Table 2, a consumer agent plans to distribute a divisible grid job of 10000 packets among the best performing providers. When first entering the agreement phase, the consumer agent selects the ranks 1 to 4 for simultaneous negotiation, totaling a maximum contracting volume of 10,243 packets. In case of unsuccessful negotiations, the consumer agent repeatedly selects provider agents to negotiate with by the above scheme until the total number of packets required by the grid job is contracted. Thus, we divide a multi-source allocation problem ("co-allocation") into a sequence of bilateral negotiations.

Rank	Factory Handle	p [packets/sec]	*r* [sec]	Factory Max Packets	Total Max Packets
1	**b**	**12.5**	**450**	**5,625**	**5,625**
2	**g**	**6.5**	**360**	**2,340**	**7,965**
3	**a**	**3.5**	**500**	**1,750**	**9,715**
4	**d**	**3.3**	**420**	**528**	**10,243**
5	**h**	**0.8**	**350**	**280**	**10,523**
6	**c**	**0.3**	**250**	**75**	**10,598**

Table 2: Example Provider Ranking in the Information Phase

5.2 Distributive Negotiation Phase

In distributive negotiations, the agents are claiming value at the expense of their opponent ("win/lose-negotiations"). The agents use a competitive negotiation strategy to decide how to divide a fixed resource, i.e. how to get the most out of a grid service contract for themselves. The result of this distributive phase is a preliminary contract that at least yields both agents' minimum utility threshold (reservation utility). Asymmetric utility distribution among two contractors indicates to what extent an agent has "won" or "lost" a negotiation. By using a more sophisticated strategy or by exploiting a weak bargaining position of their opponent, an agent may gain a bigger share of the pie.

A negotiation protocol is a transaction processing framework that contains the basic rules of the agents' interaction. SettleBot proposes a two-phase negotiation protocol. The first phase (distributive phase) supports sequential alternating offer exchange among two agents that apply heuristic strategies to gain advantage over their competitors under incomplete information, i.e. private valuation functions. The second phase (integrative phase) aims at finding joint gains while maintaining the utility distribution outcome of the first phase.

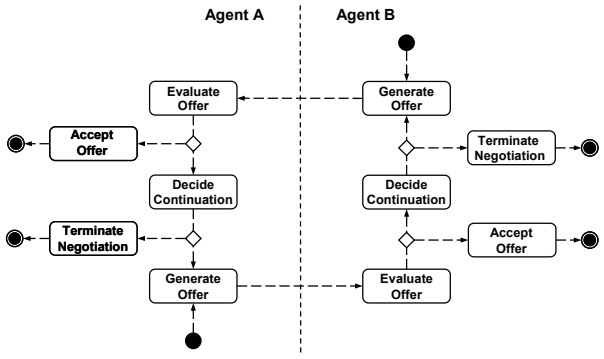


Figure 6: Negotiation Protocol (Distributive Phase)

Figure 6 depicts the sequence of decisions made by the negotiating agents in the distributive phase. A consumer agent A that has selected a matching provider B initializes a distributive negotiation by generating an initial offer $x^{A\to B,t_0} = (x_1^{t_0}, x_2^{t_0}, ..., x_n^{t_0})$. In subsequent rounds, the initial attribute values $x_i^{t_0}$ are modified by an offer modification function $\Delta(\boldsymbol{\pi})$ that represents an agent's concession strategy. An agent's strategy takes into account the agent's knowledge about the user's goals (e.g. attribute weights), knowledge about the market environment (e.g. supply/demand ratio) and knowledge about the opponent (e.g. its Quality of Service-rating). All the knowledge an agent acquires by reading its sensors, by preference elicitation and by communicating with the application interface is summed up as its belief about the world state $\boldsymbol{\pi}$. If the world state changes during an ongoing negotiation, the agent adapts its strategy accordingly, e.g. increases or reduces its concession rate. The modification function $\Delta_i(\boldsymbol{\pi})$ has the range [0;1] and thus $\Delta_i(\boldsymbol{\pi})\cdot\varphi_i$ gives the proportion of the maximum concession $\varphi_i = a_i - r_i$ currently considered strategically appropriate by the agent. Hereinafter, all functions referring to $\boldsymbol{\pi}$ are subject to knowledge processing optimization by genetic optimization (see below).

A modified attribute value offered at time *t* is given by

$$x_i^t(\boldsymbol{\pi}) = x_i^{t_0} + \Delta_i(\boldsymbol{\pi})\cdot\varphi_i . \tag{2}$$

The set of modification functions

$$\Delta_i(\boldsymbol{\pi}) = \left(\frac{t - t_0}{t_{crit}(\boldsymbol{\pi})}\right)^{\beta_i(\boldsymbol{\pi})} \tag{3}$$

represents aggressive (or "Boulware"), neutral and defensive (or "Conceder") concession strategies. We agree with Faratin, Sierra and Jennings on the suitability of such strategy types in terms of individual utility maximization and allocation efficiency [Fara+98]. The factor $\beta_i(\boldsymbol{\pi})$ determines the curvature of the modification function and thus the strategy type. Figure 7 shows the outcome of a negotiation for a grid service where the agents apply different modification strategies. In the example given by Figure 7, the nearly linear (neutral) modification strategy of the consumer agent yields higher utility than the aggressive strategy applied by the opposing provider agent.

$t_{crit}(\boldsymbol{\pi})$ is an agent's time limit for a negotiation. If a negotiation of two agents A and B starts at time $t_0^{A\leftrightarrow B}$, an agent A reaches its maximum concession at time $t_0^{A\leftrightarrow B} + t_{crit}^{A}(\boldsymbol{\pi})$. If time $t_0^{A\leftrightarrow B} + t_{crit}^{A}(\boldsymbol{\pi})$ has elapsed and agent A gains no more utility increase from new offers submitted by agent B, it terminates the negotiation. Like $\beta_i(\boldsymbol{\pi})$, the world state dependent strategy control variable $t_{crit}(\boldsymbol{\pi})$ determines an agent's negotiation stance (aggressive, neutral, defensive). An agent with a rela-

tively short $t_{crit}(\pi)$ shows defensive behavior by quickly spending its concession contingent to avoid termination by the other party.

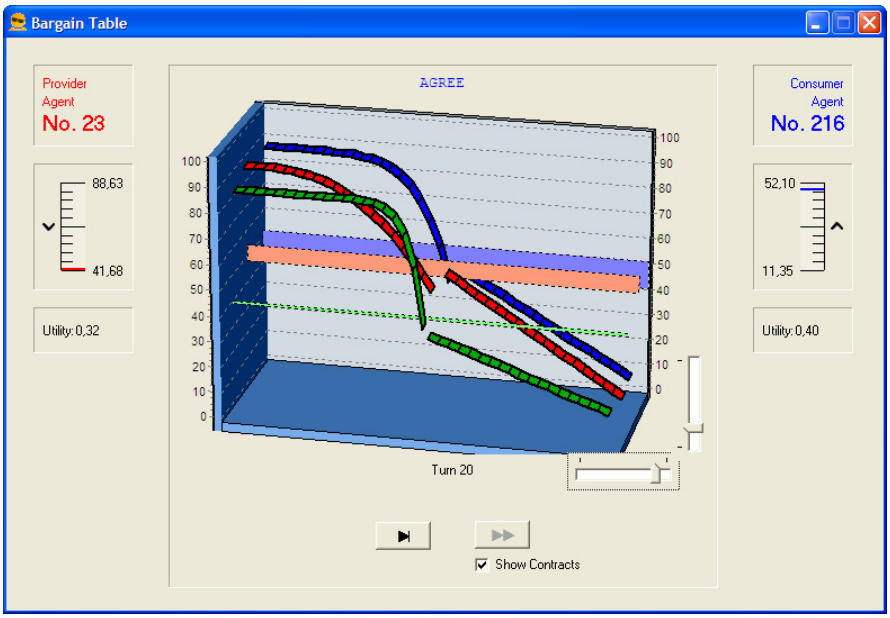


Figure 7: Negotiating Agents' Modification Strategies

The agent's utility function is designed such that $u^{A}(a_1^{A}, a_2^{A}, ..., a_3^{A}) = 1$ and $u^{A}(r_1^{A}, r_2^{A}, ..., r_3^{A}) = 0$. An important element of a negotiating agent's strategy is at what utility level a counteroffer is acceptable. Therefore, an agent determines its reservation utility u_{target} against the current world state, using rules found by test runs with a genetic algorithm. Preferably, this threshold states a value such that an offer is not accepted if in succeeding negotiations a better result can be attained. A high u_{target} increases the risk of termination without reaching agreement, while a low threshold value may waste utility which could have been attained. Along with other strategy control functions, the mapping function $u_{target}(\pi)$ is subject to genetic optimization using the GeneLab simulation environment.

The GeneLab platform simulates an agent based commercial grid and allows finding heuristic negotiation strategies by a genetic algorithm (GA). There are two major fields of explorative simulations conducted with GeneLab, the search for static and for dynamic strategies. The success of a static strategy does not depend on the dynamically changing world state but only on its fit with the negotiation mechanism. The SettleBot project focuses the search for dynamic strategies that change the agent's behavior depending on its perception of the world state. The

agent's view of the world state is derived by processing sensor readings at runtime. An agent's decision core links its strategy control variables to its sensor readings by mapping rules that are genetically optimized using GeneLab. To get more representative fitness values, the GeneLab GA uses a fitness function that calculates agent fitness as the average utility an agent gains from several negotiations. The floating point numbers representing an agent's strategy control variables are encoded as Gray code [Pres+92] to improve numerical optimization. Compared to regular binary coding, Gray-encoded numerical parameters behave smoothly under bit mutation, thereby allowing a local stochastic search for fitness maxima. The agents entering the mating pool are picked from the population by tournament selection [GoDe91].

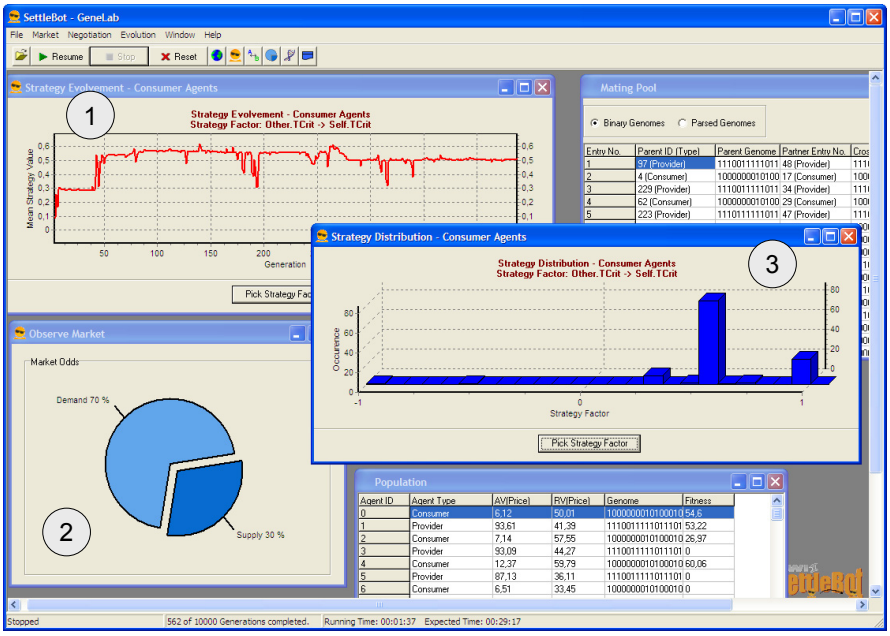


Figure 8: SettleBot GeneLab: Exploring Negotiation Strategies for the Commercial Grid

Figure 8 shows a simple example of a test run with a population of 250 agents that aims at the optimization of the function $t_{crit}(\boldsymbol{\pi})$ that determines an agent's critical time limit depending on its world state perception $\boldsymbol{\pi}$, specifically $\boldsymbol{\pi}_{other.tcrit}$, which is the average time limit of the other negotiating party as observed by the agent. The test run served to prove the suspected interdependence between an agent's market power in terms of current supply and demand ratio and its defensiveness in terms of t_{crit} as a strategy control variable. Chart 1 of Figure 8 shows an approximate evolutionary trend to $t_{crit}(\boldsymbol{\pi}) = \boldsymbol{\pi}_{other.tcrit} \cdot 0.5$ for consumer agents when the

market power ratio is 7:3 in favor of the supply side (see Chart 2). The frequency distribution depicted by Chart 3 indicates a strong fitness dominance of consumer agents adjusting their t_{crit} to half of that of their opponent. Chart 3 shows that almost the entire population has set their corresponding strategy factor to ≈ 0.5 after 562 Generations.

Other than $t_{crit}(\pi)$ and $u_{target}(\pi)$, optimizations done with GeneLab's GA engine include finding dominant mapping functions $r_i(\pi)$ and $\beta_i(\pi)$. The idea behind dynamic reservation values $r_i(\pi)$ and dynamic concession stance $\beta_i(\pi)$ is that rational agents are expected to concede more on less important attributes and vice versa. Several test runs aimed at analyzing advantageous processing rules that optimize $r_i(\pi)$ (strategy example given below) and $\beta_i(\pi)$ depending on the attribute weights w_i. Several simulations and scripted test runs in dynamically changing negotiation settings proved that successful agents improve their average utility by proportionally adjusting their $r_i(\pi)$ to the attribute weights. Our simulation results can be generalized by the strategy

$$r_i(\pi) = r_i + (a_i - r_i) \cdot \varepsilon \cdot w_i \cdot n \text{ with } \varepsilon \in \mathbb{R}^+ \text{ such that} \tag{4}$$

$$\sum_{i=1}^{n} w_i \cdot u_i \left(r_i + (a_i - r_i) \cdot \varepsilon \cdot w_i \cdot n \right) = u_{target}(\pi) \tag{5}$$

Thereby, the reservation value r_i of each attribute i is shifted towards the aspiration value a_i, depending on $u_{target}(\pi)$ and the weights w_i. As a result, an agent that generates offers according to these r_i will not fall short of its currently claimed utility $u_{target}(\pi)$ and will have more restrictive reservation values for important attributes i with $w_i \cdot n > 1$, given that $\sum_{i=1}^{n} w_i = 1$. Test runs show that the above strategy performs best if the negotiating agents' attribute weight profiles differ largely from each other. Tested with random attribute weights, an agent applying the strategy gains an average utility surplus of 16% over agents that do not modify reservation values according to their attribute weights.

The result of the distributive phase is a preliminary contract that represents both agents' individual bargaining success. An agent's success varies widely depending on what sensor data is available and how this sensor data is processed to control the strategy control variables. As expected, the distributive phase performs below Pareto efficiency since utility functions are private knowledge and both agents act strictly on behalf of their own self-interest. For further improvement of the preliminary contract, the agents subsequently enter an integrative phase of their negotiations when preliminary agreement was reached.

5.3 Integrative Negotiation Phase

The integrative phase aims at enlarging the pie, i.e. finding contract modifications that result in joint utility gains for both of the agents ("win/win-negotiations"). The goal of SettleBot's two phased negotiation mechanism is to find Pareto-undominated contracts by integrative negotiation while preserving asymmetric utility gains obtained by superior strategy in the distributive phase. We argue that a rational agent with a strategy to exploit its strong bargaining position does not opt into a negotiation protocol that aims at Pareto-efficient contracts if there is an alternative protocol that allows to profit on its advantageous situation at the expense of its opponent.

In the integrative phase, the agents search for mutual improvements by exchanging offers that are modifications of the preliminary contract. Therefore, an agent randomly modifies the preliminary contract by a Gaussian distribution, such that there is a high probability of minor modification and a low probability of major modification for each of the attributes. If an agent A finds a modification that improves its utility, i.e. finds a favorable trade-off of attribute values, it sends the modification to its opponent B as a new contract proposal. If the offer is accepted by B, there's a new preliminary contract. In any case, following an alternating offer protocol, B generates the next modified contract based on the current preliminary contract until no further improvements are found. Since utility functions are private and none of the agents (except the "Auditor" agent used in our simulation, see Figure 9) knows whether the Pareto frontier has been reached, A and B need a suitable alternative termination rule. 1000 offer exchanges and no improvement has proved a feasible restriction in our simulations with the MISTRESS simulation environment for the integrative phase. Pareto-undominated contracts are usually found in well below 1000 rounds (see Figure 9).

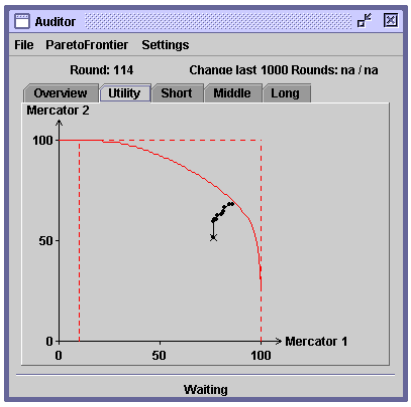


Figure 9: Approximating the Pareto Frontier in the Integrative Phase

Since modifications are random and both agents' apply neutral decision functions that accept a modification if there is a utility gain (see below), the expected total utility gain in the integrative phase is approximately equal for both agents. However, the non-deterministic nature of the integrative phase results in certain deviations from each agent's "fair utility share". In 18 percent of the cases, the deviation from the agents' fair share exceeds 10 percent. In general however, the resulting contracts maintain the utility surplus the agents achieved in the first phase, while approximating undominated contracts (points on the Pareto frontier, see Figure 9).

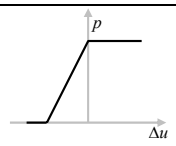
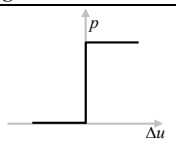
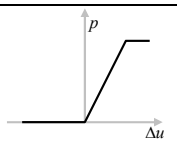
		Agent 2		
		 **Defensive**	 **Neutral**	 **Aggressive**
Agent 1	**Defensive**	**76% / 76%**	**26% / 95%**	**25% / 94%**
	Neutral	**95% / 26%**	**81% / 81%**	**83% / 78%**
	Aggressive	**94% / 25%**	**78% / 83%**	**76% / 76%**

Table 3: Utility Gains in the Integrative Phase

Table 3 shows the average utility gains of agents applying different strategies in percent of the maximum possible gain. The defensive strategy accepts a new contract with a non-zero probability even if there is a utility loss. Aggressive strategies claim a higher utility gain than neutral strategies. We found that the neutral strategy outperforms the aggressive strategy at a small margin of 83/78 due to the aggressive strategy's reluctance to accept small utility increases, thus there is no incentive to unilaterally deviate from a neutral to an aggressive strategy. Defensive strategies are inferior to all non-defensive strategies. Therefore, the neutral vs. neutral strategy pair is a mutual best response (Nash equilibrium).

6 Conclusion and Further Work

The SettleBot project extends a growing body of research that applies genetic algorithms to decision and negotiation support. While many research efforts focus decision strategies that are purely mechanism dependent, the SettleBot project aims at developing dynamic decision strategies. Dynamic strategies consider live sensor data measuring the dynamically evolving world state to reduce the downsides of incomplete information and to leverage unilateral knowledge gains for higher utility. Beginning with Axelrod's work on strategy evolution in repeated

games [Axel87], related research efforts have analyzed superior decision strategies by evolutionary computing for both game theoretic and heuristic mechanisms [Dwor+96; Gerd+03; Oliv94].

Grid and Agent technology are increasingly regarded complementary [Fost+04]. Currently, both technologies emerge as a combined field of research. Besides finding solutions for the technological requirements of grid markets [Czaj+02; Czaj+03; Fost+02], many publications focus the economic engineering of resource allocation mechanisms [Chel+04; Chen+02; Gagl+95; Keny02; ReNi98; Wols+01].

Finding applicable mechanisms for the negotiation of service level agreements among grid partners is just one of many problems to be solved when implementing even the first stage of a commercial grid (see section 1). Besides security and platform compatibility issues, successful deployment of a commercial grid requires monitoring and penalization of deviations from service level agreements. In scientific grid applications, quality of service often follows a "best effort"-approach (soft QoS-constraints). Since "best effort-service" has near-zero economic value [Chel+04], there must be hard guarantees on quality of service in a commercial grid. Therefore, several research efforts aim at developing grid architectures that allow monitoring of hard QoS-constraints agreed upon by the contracting parties. We propose a reputation mechanism to enforce the contractors' compliance with the service level agreement. In the SettleBot system, compliance data may be logged and published by the grid carrier agent. Negotiating agents may query the grid carrier agent for another agent's QoS-ratio q, i.e. the percentage of past transactions properly settled. In the information phase, another agent's QoS-ratio may serve as an additional ranking/selection-criterion. For example, a consumer agent A ranks a provider agent B by its *expected* performance $E(p^{\mathrm{B}}) = q^{\mathrm{B}} \cdot p^{\mathrm{B}}$ [packets/sec]. In the agreement phase, we expect q^{B} to have a distinct impact on a rational agent's behavior when presented to the agent as another knowledge bit forming input to its negotiation strategy.

As yet, we do not apply dynamic strategies in the integrative phase. However, we are working on a model based on a probabilistic Bayes tree that allows an agent to estimate its opponent's goals by analyzing another agent's behavior in the distributive phase. The resulting estimated goal model allows improved offer generation in both the distributive and the integrative phase.

Our future overall research goal is to propose a generic knowledge driven strategy for the commercial grid. To achieve this, we will analyze the strategic implications of information availability (reputation data, forecasts, trade history, opponent log) and its impact on welfare and individual success in the grid economy.

References

[Axel87] Axelrod, R.: The Evolution of Strategies in the Iterated Prisoner's Dilemma, in: Genetic Algorithms and Simulated Annealing, Morgan Kaufman Publishers 1987.

[Chel+04] Cheliotis, G.; Kenyon, C.; Buyya, R.: 10 Lessons from Finance for Commercial Sharing of IT Resources, in: Peer-to-Peer Computing: The Evolution of a Disruptive Technology, IRM Press 2004.

[Chen+02] Chen, C.; Maheswaran, M.; Toulouse, M.: Supporting Co-Allocation in an Auctioning-Based Resource Allocator for Grid Systems, in: Proc. of the Int. Parallel and Distributed Processing Symposium, IEEE Press 2002.

[Czaj+99] Czajkowski, K.; Foster, I.; Kesselman, C.: Resource Co-Allocation in Computational Grids, in: Proc. of the 8th IEEE Int. Symposium on High Performance Distributed Computing, IEEE Press 1999.

[Czaj+02] Czajkowski, K.; Foster, I.; Kesselman, C.; Sander, V.; Tuecke, S.: SNAP: A Protocol for Negotiating Service Level Agreements and Coordinating Resource Management in Distributed Systems, in: Lecture Notes in Computer Science (2537), Springer 2002.

[Czaj+03] Czajkowski, K.; Dan, A.; Rofrano, J.; Tuecke, S.; Xu, M.: Agreement-based Grid Service Management (OGSI-Agreement), Presentation at Global Grid Forum 2003, 2003-06-12,
Online: http://www.globus.org/research/papers/ OGSI_Agreement_2003_06_12.pdf

[Dero+03] De Roure, D.; Baker, M. A.; Jennings, N. R.; Shadbolt, N.R.: The Evolution of the Grid, in: Grid Computing: Making the Global Infrastructure a Reality, Wiley 2003.

[Dwor+96] Dworman, G.; Kimbrough, S.; Laing, J.: Bargaining by Artificial Agents in Two Coalition Games: A Study in Genetic Programming for Electronic Commerce, in: Proc. of the AAAI Genetic Programming Conference, Stanford Bookstore 1996.

[Fara+98] Faratin, P.; Sierra, C.; Jennings N. R.: Negotiation Decision Functions for Autonomous Agents, in: Int. Journal of Robotics and Autonomous Systems 24 (3-4), Elsevier 1998.

[Fost+02] Foster, I.; Kesselman, C.; Nick, J.; Tuecke, S.: The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, in: GGF 2002, Online: http://www.globus.org/research/papers/ogsa.pdf.

[Fost+04] Foster, I.; Jennings, N. R.; Kesselman, C.: Brain meets brawn: Why Grid and Agents Need Each Other, in: Proc. of the 3rd Int. Conf. on Autonomous Agents and Multi-Agent Systems, (to appear) 2004.

[Gagl+95] Gagliano, R. A.; Fraser, M. D.; Schaefer M. E.: Auction allocation of computing resources, in: Communications of the ACM (38) 6, ACM Press 1995.

[Gerd+03] Gerding, E.; Van Bragt, D.; La Poutre, H.: Multi-Issue Negotiation Processes by Evolutionary Simulation, Validation and Social Extensions, in: Computational Economics (22), Kluwer Academic Publishers 2003.

[GoDe91] Goldberg, D. E.; Deb, K.: A Comparative Analysis of Selection Schemes Used in Genetic Algorithms, in: Foundations of Genetic Algorithms, Morgan Kaufmann 1991.

[KeRa93] Keeney, R. L.; Raiffa, H.: Decisions with Multiple Objectives: Preferences and Value Tradeoffs, Cambridge University Press 1993.

[Keny02] Kenyon, C.: Grid Economics: Grid Value and Practical Realization, Presentation at CERN CC, 2002-09-27,
Online: http://www.zurich.ibm.com/pdf/ GridEconomics/Grid_Value_Basis_(long)_ CERN270902.pdf

[Kran+71] Krantz, D. H.; Luce, R. D.; Suppes, P.; Tversky, A.: Foundations of Measurement, New York Academic Press 1971.

[Lasz+03] Laszewski, G.; Pieper, G.; Wagstrom, P.: Gestalt of the Grid, in: Performance Evaluation and Characterization of Parallel and Distributed Computing Tools, Wiley 2003

[Oliv94] Oliver, J. R.: On Artificial Agents for Negotiation in Electronic Commerce, PhD Thesis, Wharton School of the University of Pennsylvania 1994.

[Pres+92] Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P.: Numerical Recipes in C, Cambridge University Press 1992.

[ReNi98] Regev, O.; Nisan, N.: The POPCORN Market – Online Markets for Computational Resources, in: Proc. of the 1st Int. Conf. On Information and Computation Economies, ACM Press 1998.

[Saat92] Saaty, T. L.: Multicriteria Decision Making - The Analytic Hierarchy Process, RWS Publications 1992.

[Suth68] Sutherland, I. E.: A futures market in computer time, in: Communications of the ACM (11) 6, ACM Press 1968.

[WiEd86] Von Winterfelt, D.; Edwards, W.: Decision Analysis and Behavioral Research, Cambridge University Press 1986.

[Wols+01] Wolski, R.; Plank, J.S.; Brevik, J.; Bryan, T.: Analyzing Market-based Resource Allocation Strategies for the Computational Grid, in: Int. Journal of High Performance Computing Applications (15) 3, Sage Publications 2001.

[Zbor96] Zbornik, S.: Elektronische Märkte, elektronische Hierarchien und elektronische Netzwerke, Universitätsverlag Konstanz 1996.