Catallaxy-based Grid Markets

Torsten Eymann[1], Oscar Ardaiz[3], Michele Catalano[5], Pablo Chacin[3], Isaac Chao[3], Felix Freitag[3], Mauro Gallegati[5], Gianfranco Giulioni[5], Liviu Joita[4], Leandro Navarro[3], Dirk G. Neumann[2], Omer Rana[4], Michael Reinicke[1], Ruben Carvajal Schiaffino[6], Björn Schnizler[2], Werner Streitberger[1], Daniel Veit[2], Floriano Zini[6]

[1] Chair for Information Systems, University of Bayreuth, Germany
{eymann|reinicke|streiberger}@uni-bayreuth.de
[2] Chair for Information Management and Systems, University of Karlsruhe (TH), Germany
{neumann|schnizler|veit}@iw.uka.de
[3] Department of Computer Architecture, Technical University of Catalonia, Spain
{pchacin|ichao|oscar|felix|leandro}@ac.upc.es
[4] School of Computer Science and the Welsh eScience Centre, University of Cardiff, UK
{l.joita|o.f.rana}@cs.cardiff.ac.uk
[5] Department for Economics, University Ancona, Italy
{catalona|gallegati|giulioni}@deanovell.unian.it
[6] Automated Reasoning Systems Division, ITC Trento, Italy
{carvajal|zini}@itc.it

Abstract. Grid computing has recently become an important paradigm for managing computationally demanding applications (composed of a collection of services). Providing dynamic discovery of services, and the determination of a service with the "best value" remains a complex problem in Grid computing. Such a decision is a multi-attribute $n : m$ allocation problem, and often made using a centralized component in existing systems. We propose a two layered architecture for service discovery in such Application Layer Networks (ALN) and Grids. The first layer consists of a service market in which complex services are traded, and can be chosen based on price and availability. The second layer provides an allocation of services to appropriate resources in order to carry out the specific services. This work comprises the foundations for the comparison of centralized and decentralized market mechanisms for allocation of services and resources in ALN and Grid markets.

This work has partially been funded by EU IST progamme under grant 003769 "CATNETS".

1 Introduction

An investigation of implementing an electronic Grid market based on the "Catallaxy" concept of F.A. von Hayek is presented. This market makes use of a "free market" economic self-organisation approach in the context of electronic services. A key issue is to utilize this approach as the basis for realizing resource allocation in Application Layer Networks (ALNs). The term of ALN integrates different overlay networks, such as Grid and Peer-2-Peer (P2P) systems, as application layer networks above existing physical topologies of the Internet. A key theme in this work is to move away from the presence of centralized resource brokers, a common approach of many other Grid market approaches, and instead focus on the ability to trade electronic services based on decentralized mechanisms. We envision the existence of two types of markets: (1) a resource market – which involves the trading of computational and data resources, such as processors, memory, etc, and (2) a service market – which involves the trading of application services. This distinction between a "resource" and a "service" is necessary to allow different instances of the same service to be hosted on different resources. It also enables a given service to be priced based on the particular resource capabilities that are being made available by some hosting environment.

We focus on the ability of a free-market economy to adjudicate and satisfy the needs of the human agents' behaviour towards a decentralized organisation principle, where human agents are self-organised and follow their own interest. The Catallaxy approach is a coordination mechanism for systems consisting of such autonomous decentralized agents, and based on constant negotiation and price signalling between agents [1]. Catallaxy is a way to inform the individual (agent) about the knowledge of others, and provide an exchange of information that lead to the generation of prices which comply with the value every individual (agent) assigns to the respective information [4]. Catallaxy develops self-organizing individuals (agents) that are highly dynamic, thereby leading to systems which behave in a Peer-2-Peer fashion.

2 What is Catallaxy

Friedrich August von Hayek [6], and other Neo-Austrian economists understood the market as a decentralized coordination mechanism, as opposed to a centralized command economy. Apart from political macroeconomic thoughts, Hayek's work also provides concrete insight on the working mechanisms of economic coordination. The emergence of software agent technology and increasing size of information systems leads to the possibility of implementing Hayek's Catallaxy concept and using the ensuing "spontaneous order" as a concrete proposal for both the design and coordination of information systems. However, a formal description of this self-organizing market mechanism does not so far exist. The Catallaxy concept is based on the explicit assumption of self-interested actions of the participants, who try to maximize their own utility and choose their actions under incomplete information and bounded rationality [21]. The term Catallaxy comes from the Greek word "katallatein", which means, "to barter" and at the same time, "to join a community." The goal of Catallaxy is to arrive at a state of coordinated actions, through the bartering and communicating of members, to achieve a community goal that no single user has planned for. The main characteristics of the Catallaxy [7] are enumerate below. Each imposes several requirements upon the design of an information system embodying a catallactic approach.

1. Participants work for their own interest to gain income. Every system element is a utility maximizing entity, supports means to measure and compare income and utility, and to express a desire to reach a defined goal.

2. Participants can only estimate the effect of action alternatives on an income or utility maximization goal. There is no completely rational utility maximizer with total knowledge and foresight of the environment. Instead, "constitutional ignorance" of the rationally bounded participants makes it inevitably impossible to know the exact environment state. For large and very dynamic information systems, this observation leads to a design shift. Instead of trying to overcome this limitation by central means, e.g. through synchronization of the system by introducing round-based brokerage, the focus shifts to improving the computational intelligence of the actions to decide under uncertainty, and to adapt to constantly changing signals from the outside.
3. Participants communicate using commonly accessible markets, where they barter about access to resources held by other participants. The development of prices for a specific good, relative to alternatives, and whether they are increasing or decreasing, leads buyers to look for alternative sources of procurement and thus enhances the dynamics of the market. In that view, a market is simply a communication bus; not a central optimization component, or a mechanism or a protocol.

Although Hayek's Catallaxy approach is devoted to markets with human agents, it is also deemed promising as an adequate coordination mechanism for information systems. In the following we propose a way to implement the catallaxy in a Peer-2-Peer system.

3 Catallaxy in CATNETS

This section will pick up the requirements of the Catallaxy described in section 2 above and will present fundamental components to satisfy these requirements in ALNs. We start with a decomposition of the network into two distinctive markets. Then we will present the corresponding market model to achieve Catallaxy. In the remainder of this section, we identify the functionality and components needed. Subsequently, we present a middleware and a corresponding application scenario.

3.1 Environment of Application Layer Networks

ALNs encompass heterogeneous resources in different administrative domains, which have been logically coupled together. This comprises both computational and data services. We expect ALNs to be dependent on basic services that can be dynamically combined to form value-added complex services [22]. These basic services require a set of resources, which need to be co-allocated to provide the necessary computing power (like in computational Grids). The orchestration and configuration of these basic services and resources can be understood itself as an inherent service. Such orchestration must be hidden from the application, and managed through the middleware. We thus divide the environment into two layers, the application/service layer and the resource layer. In these two layers, we contemplate three different roles, which are: (1) complex services (application layer), (2) basic services (application layer and resource layer), and (3) resources (resource layer). Basic services may also provide an interface to access computational resources for complex services. In both layers, the participants have various objectives which might change dynamically and unpredictably over time. Traditional approaches, using centralized policies require complete state information which is often not available in dynamic and complex networks [10]. As an acceptable system-wide performance matrix is impossible to define, we use an economics-based paradigm for the management or resource allocation and orchestration [3].

3.2 Market Model

Current Grid Computing architectures exhibit fairly static resource infrastructure which is connected by physically stable links. The shift to a pervasive Grid, that could exist ubiquitously, demands for a more dynamic consideration of resources and connections. Figure 1 shows our perspective on a Catallaxy-based Grid Market. The market is understood as a decentralized control mechanism for services and resources.

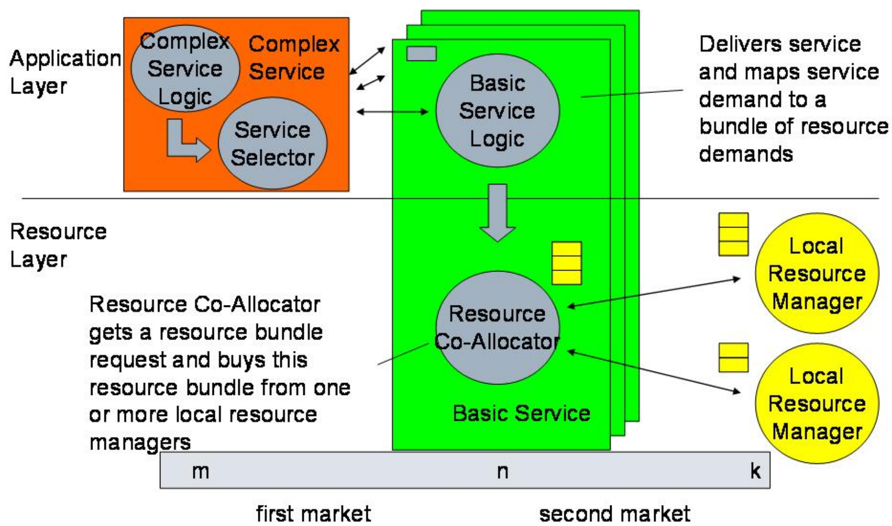

Fig. 1. Catallaxy-based Grid Market Model

A *complex service* could be represented by a proxy which needs (remote) basic service capabilities for execution – with support for a service selector instance. Complex services are therefore shielded from the details on the resource layer

A *basic service* is split into the *basic service logic* and a *resource allocator*. The logic is able to negotiate with the complex service and to translate the requirements for service execution on a resource instance (e.g. CPU, storage and quality of service requirements, etc.). A *resource allocator* gets the resource specification and broadcasts the respective demand to the *local resource managers*. This comprises bundles and co-allocative negotiations. Bundles are understood as an n-tupel of resource types (e.g.

CPU, storage, and bandwidth); co-allocation describes obtaining resources for one single service transaction from various *local resource managers* simultaneously. Local resource/job scheduling is not the focus of the project. It is expected that a *local resource manager* hides all details of the allocation.

On the first market, *complex service* and *basic service* negotiate; an agent managing a *complex service* acts as a buyer, the *basic service* agent as a seller. The same market roles can be found at the resource layer, the resource allocator is the buyer agent, the *local resource manager* acts as a seller agent.

Contemplating the second market, it is a n to k market: n basic service copies can bargain with k resource services. This takes dynamic resources into account. Resources are in our view entities that can fail like basic services which are subject to maintenance and inspection procedures or link failures.

3.3 Integration of the Markets

For offering a basic service within a Catallaxy-based Grid market, it is necessary to contract the required resources (on resource layer). This may be achieved as follows:

1. Contracting resources in advance: requires a forecast of future demand [3]. For a centralized allocation mechanism this might be suitable as demand and supply fluctuations can be absorbed over the whole network. In decentralized allocation demand and supply can change rapidly, and the decision-makers will not be able to anticipate this situation by their local knowledge. Therefore, they will be exposed to a higher risk of bankruptcy.
2. Contracting resources after closing the service contract: this might lead to insufficient resource offers on the resource market and thus to unaccomplishable contracts.
3. Contracting the resources during negotiation: in this approach, before giving a proposal several local resource managers are contacted. This has the inherent advantage that supply changes in the resource market can be transferred immediately to the service market. This reduces risks for the basic service and balances both markets.

Scenario 3 is superior to the others and will thus be incorporated in our Catallaxy-based grid market. For service execution the basic service logic requests a resource bundle (*requestResourceBundle*). The further process of contracting/allocating the resource is done by the resource co-allocator. The selection of a resource bundle is done analogous to the selection of a service, with the exception that a bundle is requested (*selectResourceBundle*), whereas on service market only one service can be negotiated per request.

The local resource managers offer resource bundles (*offerResource*). The resource bundle could be a tuple consisting of resource properties such as bandwidth, CPU, and storage (for instance). The manager is the seller agent of the resource market, having the ability to negotiate with the resource allocator (*negotiateItem*). The negotiation is also initiated by the resource co-allocator.

4 Implementation Techniques

In our Catallaxy approach, every player in the market is modelled as a software agent. First the lifecycle of agents and their components are presented, then their integration into a middleware is shown.

4.1 Lifecycle of Agents

This section presents a general model of transaction phases and the mappings of this model to the buyer and seller agents (see figure 2). The lifecycle starts with an initialisation phase – consisting of information and matching sub-phases (see [23]).

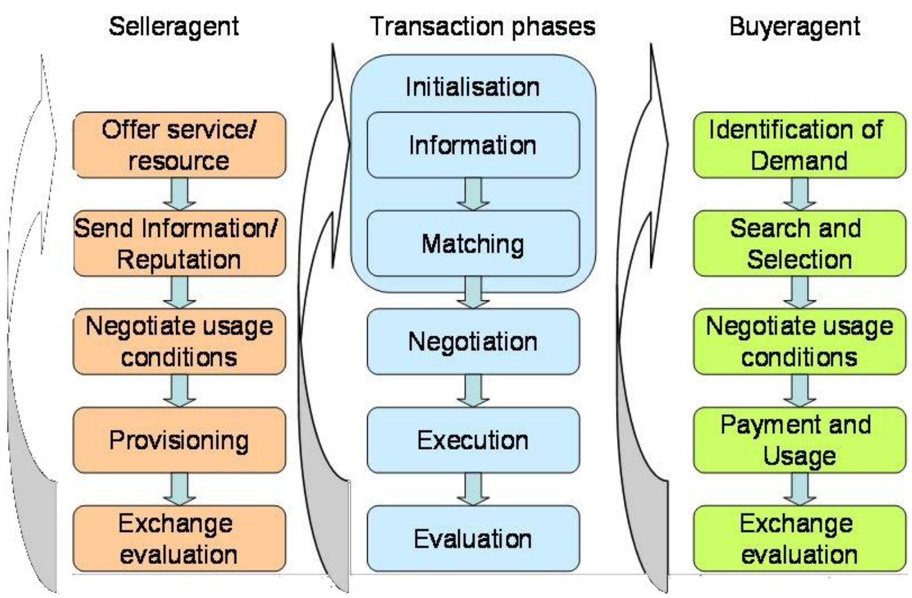

Fig. 2. Lifecycle of Agents

- In the information phase the buyer agents (resource co-allocator and complex service) try to identify demand. The seller side (basic service and local resource manager) offers items to be sold.
- The matching phase brings together the buyer and seller agent. The buyer agent initiates this process after specification of its demands. Seller agents can support the buyer agents with additional information of the service.

A parallel, bilateral exchange of information between buyer and seller agents shapes the negotiation phase. The usage conditions between seller and buyer agents are multiattributive items (like basic service on the first market and resource bundles on the second market). Subsequently, the execution phase contains deployment and clearing of the contracted service, which the seller agent delivers on demand. Often the evaluation phase is omitted and the process begins again. In CATNETS, the buyer-seller relationship is analyzed in the evaluation phase. Information about evaluation is exchanged between them.

4.2 Components for Realizing Catallaxy

For the realization of the Catallaxy paradigm several components have to be implemented in the decentralized architecture. For the preparation and calculation of price proposals, a negotiation module is required that constitutes the interface between internal perception of the environment and the surrounding (sensor and effector). These negotiation strategies need to use learning mechanisms, to react to changes in the environment and to implement a method that adapts to the behaviour of the surrounding agents.

Service discovery Discovery of suitable services is a key goal of CATNETS. Implementing a central catalogue for service discovery may not be suitable, due to the decentralized nature of the buyers and sellers. Thus, solely decentralized discovery mechanisms are considered, and involve:

Unstructured discovery: The simplest decentralized search method is using an unstructured flooding mechanism [16]. Flooding works under the assumption of a nodes' neighbour relations. Queries are not transmitted to a central catalogue, but instead distributed among the peers. A search request is forwarded to all neighbours of a peer (with a particular Time to Live (TTL) to restrict propagation).

Structured discovery: Structured search algorithms promise a guaranteed item discovery and a reduced message count. The usage of distributed hash tables (DHTs) in CHORD [2], TAPESTRY [25], VICEROY [11] offers a guaranteed search, and distributes the search process to the connected nodes in the network. The search does not rely on a random query propagation in the network, but calculates the closest known node to the requested service instance.

DHTs lack scalability in dynamic networks, as state changes (e.g. churns) lead to high overhead and might influence the simulation behaviour considerably. Thus, we regard the implementation of a simple flooding algorithm to be the best approach for CATNETS.

Negotiation As a basic principle, the negotiation strategy constitutes a search process in a space of potential agreements. The dimension of this search space is identical with the number of negotiation attributes. Thus, a negotiation comprising quality of service, delivery time, and price spans a 3-dimensional search space. In several cases, it is possible to collapse various attributes into one criteria "price", for example when delivery time affects the buyer's usage and therefore justifies a change of price. Negotiation concerns are outlined below.

Type of negotiation An automatic negotiation in an electronic market is shaped by an interaction of two or more software agents. These negotiations can be accomplished by integrative or distributive negotiation [9, 15]. In integrative negotiations, participants exchange information about objectives and priorities to seek for a common solution. This concept is recommendable if the opponents have to accept the negotiation dimensions which cannot be represented by prices. This postulates a cooperation of the opponents for reaching the agreed target. Distributive negotiations imply a participant's step-by-step acceptance of concessions, bringing both opponents closer in their expectations in every negotiation round. Distributive negotiations are marked by existence of a common utility space [15], that can be represented by a price. Thus, distributive negotiations give the option to reduce the negotiation dimensions. This should result in a zero-sum game, the utility one looses can be gained by the opponents and the global utility in the systems remains constant.

Goal The goal is a system wide pareto-optimum that can be consulted as an acceptable doctrine of general goodness [17]: A solution X is pareto-optimal if no agent can further ameliorate the achieved result without discriminating an opponent. That implies that if solution X is not pareto-optimal, both agents could negotiate a deviating solution that promises pareto-optimality. [18] extends this approach by introducing various additional criteria for the optimality determination: from game theory he uses the Nash-equilibrium that emerges if no agent has an incentive to diverge from its chosen selection. Translated to prices, this means that pareto-optimality is a state in which no agent can increase its budget without decreasing the budget of other agents (compare zero-sum game). Utility can be understood as budget increase per transaction and per period, sales volume or other values taken from business economics.

Strategy The definition of a strategy about how to reach the objectives of a negotiation is essential for modelling a market. A (human) principal defines an indifference price that equals his or her estimate of the value of the good. For a buyer, this is a maximum price, for the seller a minimum price. So the utility gain equals the amount between price of the purchase and the indifference price. The start price represents the price where the negotiation strategy begins. By agreeing concessions, the opponents come closer to the middle and a possible contract. A transaction is unlikely if the closure zone is empty, which might result when indifference prices do not build an overlapping zone.

The bargaining protocol can be implemented in different modes:

1. Buyers and sellers give their start prices without agreeing concessions. Thus, a contract can exclusively be accomplished, when the start price of one participant is already in the closure zone. An example is the usage of catalogues, where offer prices are fixed.
2. Only the seller performs concessions and the buyer remains at its start price. This is represented by the Dutch auction.
3. Only the buyer performs concessions and the seller remains at its start price. This is represented by the English auction.

4. Both agents get closer each negotiation step. This sequence of concessions describes a double auction [5].

However, an agent could reject the proposal, accept it or send a counter-offer. Various alternative and approaches are possible to handle these scenarios.

4.3 Middleware Implementation

The Catallactic middleware has been envisioned as a set of economic agents that interact with each other, and the software components of the underlying ALN. This acts as a coordination technique and makes use of economic criteria for the assignment of resources, as can be seen in the figure 3.

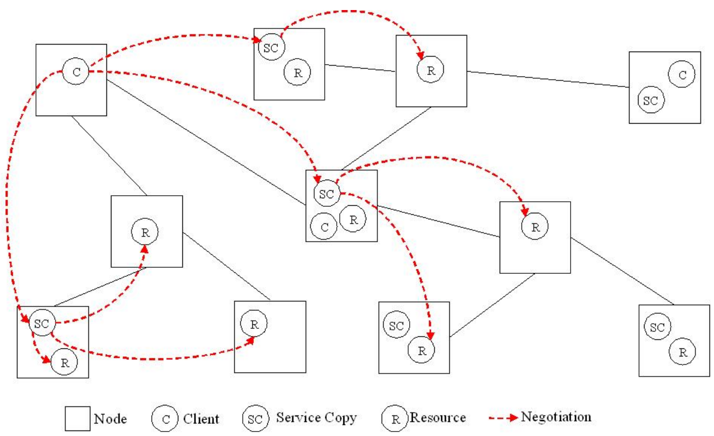

Fig. 3. Catallactic middleware as a network of agents

Such economic agents interact under a P2P architecture. The term P2P should be interpreted not as an specific system architecture, but as a general approach for distributed system design that can be realized under very different architectures and topologies, ranging from unstructured distributed networks to very centralized systems [12]. P2P systems exhibit a set of characteristics that are relevant from the architectural point of view [13]:

- Decentralization: there is no single or centralized coordination or administration point.
- Symmetric interaction between peers: all peers are simultaneously clients and servers requesting service of, and providing service to, their network peers.
- Non-deterministic topology: At any moment in time, the set of member nodes and overall topology of the network is unpredictable.

– Heterogeneity: The devices contributing in applications can differ in many properties as resources, performance and trustworthiness.
– Communication paths between peers are created dynamically based on various factors, like network conjunction or intermediate peers state.

These characteristics, when considered together, lead to a set of stringent architectural requirements for self-organization. The dynamic nature of the network prevents an apriori configuration of the peers, or the maintenance of centralized configuration files. The peers need to discover continuously the network characteristics and adapt accordingly. This requires a distribution of some important system functions like resource and topology management, traditionally reserved to specialized nodes.

Instead of implementing Catallactic agents responsible for both the self-organization of the system and the management of the negotiation process, we propose a layered architecture that is shown in the figure 4. In this architecture, economic agents are responsible for implementing the high level economic behavior (negotiation, learning, adaptation to environment signals, strategies of other agents). Application services delegate activities such as negotiation to the economic agents. Economic agents rely on a lower P2P Agent layer for the self-organization of the system, and the interaction with the base platform that ultimately manages the resources being traded.

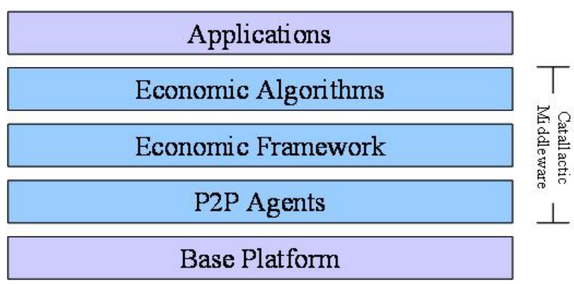

Fig. 4. Layered Agent Architecture

This architectural approach offers the direct benefit of a clear separation of concerns between the layers, which beside helping in tackling the complexity of the system, also facilitate the construction of a more adaptable system as the upper layers can be progressively more specialized (by means of rules and strategies used in the negotiations) into specific application domains.

However the separation of economic strategies and policies from the enforcing mechanisms introduces some important design and implementation issues:

– How the description of the service requirements along with the desired conditions (preferences) are passed by the application to the economic agents and how any missing information that can not be provided by the application could be automatically filled. One example of such information is the application's budget to negotiate for resources.

- How the self-organization layer can adapt its configuration and behavior to the results of the economic negotiation. For example, the adjustment of the distributed search for resources to extend or contract its scope based on the economic outcomes of the agent (if the agent is not obtaining acceptable outcomes or is not fulfilling its requests, the search scope should be extended to include more peers in the agent's market).
- How to enforce system wide rules for markets without appealing to centralized institutions. These rules are needed to offer participant agents a certain level of confidence about the fulfillment of agreed conditions and service level agreements.

We believe these requirements demand an innovative approach for middleware construction based on some general design principles:

- Create a general middleware framework, which defines the over-all architecture and offers a set of generic mechanism, on which specialized mechanism can be dynamically plugged to adjust to specific application domains or market design.
- Support a two way communication between layers (instead of the traditional unidirectional communication from top to bottom), to allow lower layers inform upper layers of relevant events. Upper layers will be able to update their strategies and pass updated policy parameters back to lower layers.
- Make information about the system behavior readily available to economic agents, gathering and disseminating it from the middleware framework so that information sharing do not depend entirely on the agents themselves, but will come from the "market environment". This information will be limited, however, to the externally observable properties, as the number of negotiations and the success or failure of negotiation. Agent's internal information will not be accessible unless the agent itself makes it available.

4.4 Application Scenario

To implement a scenario demonstrating the use of the Catallaxy approach, we propose the architecture illustrated in figure 5. The figure demonstrates the use of various components identified in section 3 for sharing content. The architecture consists of a number of Site Monitors (SM) with a number of Master Grid Services (MGS) under their control, each of those MGS having a cluster of Grid computers acting as slaves that perform the received jobs.

Site Monitors undertake an important role within the system, and are responsible for:

- Establishing and maintaining a P2P communications infrastructure.
- The point of propagation of "call for bids" market messages between Grids/P2P sites and other peers. An example is the use of an auction protocol, where such site monitors act as auctioneers for their own sites. The site monitor therefore acts as a control authority for a particular site participating in the market, as well as nodes in a P2P/Grid topology.
- They also act as a rendezvous point in a P2P topology – essentially supporting the caching of messages that are propagated in the network.

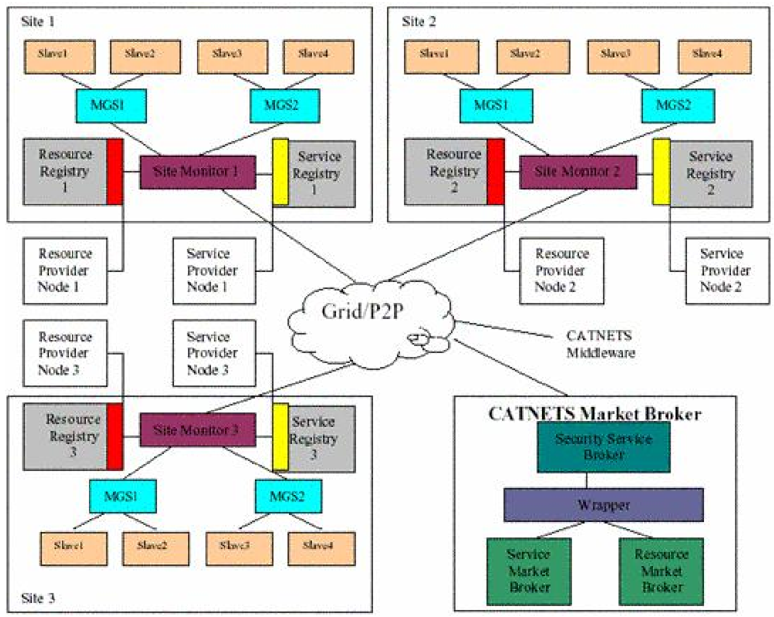

Fig. 5. The Proposed Multiagent System Architecture for Program Indexing and Querying

– They can also provide a service or resource registries – responsible for registering all services available within their site.

A resource provider node i would be responsible for matching the requests from market resource agents with the available resources on site. Similarly, a service provider node j would be responsible for matching the requests from market service agents with the available services on site. Both resource and service provider nodes will be agent based nodes capable of hosting agents that interact within a "Catnets" market. Another function of these nodes will be to send notification messages, such as forwarding requests for resources/services to their neighbouring nodes. They may negotiate directly with the nearest nodes for resources/services.

5 Reference Scenarios

The decentralized negotiation protocols following the Catallaxy paradigm will we compared with centralized auction protocols. As benchmark we introduce two extreme scenarios. One scenario is characterized by fully standardized goods, whereas the other allows for highly heterogeneous goods.

For a reasonable benchmark, we have to find adequate auction protocols for these scenarios. Unfortunately, the environment and the underlying auction protocol exerts crucial effects on the outcome [8]. For instance, in a sealed bid auction the bidders simultaneously submit bids to the auctioneer without knowledge of the amount bid by other participants. In contrast, all bids under an open cry auction are available for everyone to see. Thus, in a sealed bid auction the participants do not learn as much about the valuations of the other participants as in an open cry auction. The higher information feedback may affect the bidding behavior of the market participants and could therefore lead to different outcomes. As such, designing market mechanism that achieve a desired outcome is extremely difficult, because it entails the anticipation of agent behavior.

The Market Engineering approach guides the holistic design of tailored market mechanisms by providing a structured, systematic, and theoretically profound design procedure [24]. The approach provides a process model which is divided into four stages:

In the first stage – the environmental analysis – the requirements of the new market mechanism (i.e. who are the potential participants, what are their preferences, endowments, and constraints?) are deduced. On base of the requirements, a market mechanism is designed and implemented in the second stage. Having implemented the appropriate market mechanism, it is tested upon its economic properties and its operational functionality in the third stage and finally introduced within the fourth stage.

5.1 An Application for the Service Market

Applying the Market Engineering approach to the service market, the environment has to be analyzed in the first step. Subsequently, the corresponding requirements have to be extracted.

The environment comprises the market participants. In the case of the service market, we use rather abstract descriptions of the participants. Basically, buyers and sellers are services, which require other auxiliary services. That is, we distinguish basic services as sellers (e.g. a PDF creator service) and complex services as buyers (e.g. agents requesting a specific service). The basic services offer one or more specific auxiliary services. Hence, they are responsible for providing the auxiliary services to the buyers as well as for acquiring the required resources for the services on the resource market.

Obviously, the products traded on the service market are completely standardized. For example, an instance of a PDF creator traded once does not differ from a PDF creator instance traded at a later time.

Based upon the environment definition, the requirements for a market mechanism can be summarized as follows:

Simultaneous trading The mechanism requires that multiple sellers and multiple buyers can trade simultaneously.

Immediate execution It requires that suitable buyer orders are executed immediately against suitable seller orders.

No partial execution It requires that orders are not partially executed.

Following these requirements, a continuous double auction fits these requirements [5] and serves as a comparable mechanism for the decentralized negotiation schema [20].

5.2 An Application for the Resource Market

In a resource market, participants are the basic services as resource consumers (buyers) and resource services (sellers) offering computer resources. The transaction objects are computational resources with specific capacity, e.g. processing power. Capacity is allocated based on time slots, and the same resources (e.g. CPUs) can differ in their quality attributes, e.g. a hard disk can have 30GB or 200GB of space. The requirements for the resource market are the followings ones [19]:

Simultaneous trading In analogy to the service market, the mechanism has to support simultaneous trading of multiple buyers and sellers, as well as an immediate resource allocation.

Bundle orders The mechanism has to support bundle orders – i.e. all-or-nothing orders on multiple resources – as basic services usually demand a combination of computer resources[1].

Multi-attribute orders For comprising the different capacities of the resources (i.e. resources can differ in their quality), the mechanism has to support bids on multi-attribute resources.

Reviewing the requirements and surveying the literature, no classical auction mechanism is directly applicable to the resource market. Instead, there is a need for a multi-attribute combinatorial exchange that satisfies the described requirements.

6 Conclusion

In this paper we introduced the basic concepts for the comparison of centralized versus decentralized market mechanisms in ALN. Firstly, we introduced the Catallaxy by F.A. von Hayek as a basic principle for a decentralized market approach. This approach is then translated into the decentralized market model for the CATNETS project.

In order to provide a benchmark, the decentralized approach is compared to a centrally defined market mechansim. The latter is deduced applying a structured Market Engineering approach. Furthermore, the foundation for the implementation techniques and the middleware are layered in order to achieve comparable results from both approaches in the future. The work is comprised by reference and application scenarios.

Future work in our research includes the full implementation of both market approaches and a profound evaluation of the results of both markets. Critical questions are the scalability of market mechanisms, the allocation efficiency under constraints of the number of participating entities. These and further issues are measured using appropriate metrics from economics and computer science.

[1] This is based on the fact that computer resources (e.g. in the Computational Grid) are complementarities. Complementarities are resources with superadditive valuations ($v(A) + v(B) \leq v(AB)$), as the sum of the valuations for the single resources is less than the valuation for the whole bundle.

References

1. Project deliverables, “D3: Catallaxy Evaluation Report”, March 2003. Available at: `http://research.ac.upc.es/catnet/pubs/D3.pdf`
2. H. Balakrishnan, M. F. Kaashoek, et al., “Looking Up Data In P2P Systems”, Communications of the ACM 46 (2), pp. 43-48, 2003
3. R. Buyya, “Economic-based Distributed Resource Management and Scheduling for Grid Computing”, Monash University, Melbourne, Australia, Ph.D., 2002
4. T. Eymann, M. Reinicke, O. Ardaiz, P. Artigas, L. Dmaz de Cerio, F. Freitag, R. Messeguer, L. Navarro, D. Royo, “Decentralized vs. Centralized Economic Coordination of Resource Allocation in Grids”, 1^{st} European Across Grids Conference February, 13th-14th, 2003, Universidad de Santiago de Compostela, Spain (ACROSSGRID 2003), February, 2003
5. D. Friedman, “The Double Auction Market Institution: A Survey”, in D. Friedman, J. Rust (eds), “The Double Auction Market – Institutions, Theories, and Evidence”, Cambridge MA, Perseus Publishing, 1991
6. F. A. Hayek, W. Bartley, P. Klein, B. Caldwell, “The collected works of F. A. Hayek”, University of Chicago Press, 1989
7. E. Hoppmann, “Unwissenheit, Wirtschaftsordnung und Staatsgewalt”, in V. Vanberg (eds), “Freiheit, Wettbewerb und Wirtschaftsordnung”, Haufe Verlag, 1999
8. M. Jackson, “Mechanism Theory”, Encyclopedia of Life Support Systems (EOLSS), Eolss Publishers, 2002
9. N. R. Jennings, P. Faratin, et al., “Automated negotiation: prospects, methods and challenges”, International Journal of Group Decision and Negotiation 10 (2), 2001
10. K. Krauter, R. Buyya, et al, “A taxonomy and survey of grid resource management systems for distributed computing”, Software – Practice And Experience 32 (2), pp. 135-164, 2001
11. D. Malkhi, M. Naor, et al., “Viceroy: A scalable and dynamic emulation of the butterfly”. 21st annual ACM symposium on Principles of distributed computing, ACM Press, 2002
12. D. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja1, J. Pruyne, B. Richard, S. Rollins, Z. Xu “Peer-to-Peer Computing”, HP Laboratories Palo Alto, Reseach report PL-2002-57, 2002
13. Pepito Project, “Deliverable D1.1 - Required foundations for peer-to-peer systems”, IST-2001-33234, 2003
14. W. H. Press, S. A. Teukolsky, et al., “Numerical Recipes in C++ – The Art of Scientific Computing”, Cambridge, MA, Cambridge University Press, 2002
15. D. G. Pruitt, “Negotiation behavior”, New York, Academic Press, 1981
16. M. Ripeanu, “Peer-to-Peer Architecture Case Study: Gnutella Network”, Chicago, University of Chicago, 2001
17. J. S. Rosenschein, G. Zlotkin, “Rules of encounter – designing conventions for automated negotiation among computers”, Cambridge, MIT Press, 1994
18. T. W. Sandholm, “Negotiation Among Self-Interested Computationally Limited Agents”, Amherst, University of Massachusetts, 1996
19. B. Schnizler, D. Neumann, C. Weinhardt, “Resource Allocation in Computational Grids - A Market Engineering Approach”, Proceedings of the WeB 2004, Washington, 2004
20. B. Schnizler, D. Neumann, D. Veit, M. Napoletano, M. Catalano, M. Gallegati, M. Reinicke, W. Streitberger, T. Eymann, “D1.1: Environmental Analysis for Application Layer Networks”, CATNETS Project Deliverable, 2005
21. H. A. Simon, “Models of Man - Social and Rational”, John Wiley and Sons, 1957
22. M. P. Singh, M. N. Huhns, “Service-Oriented Computing: Semantics, Processes, Agents”, New Jersey, John Wiley and Sons, 2005
23. D. Veit, “Matchmaking in Electronic Markets - An Agent-based Approach towards Matchmaking in Electronic Negotiations”, LNAI 2882, Springer Heidelberg, 2003
24. C. Weinhardt, C. Holtmann, D. Neumann, “Market Engineering”, Wirtschaftsinformatik 45 (6), pp. 635–640, 2003
25. B. Zhao, J. Kubiatowicz, et al., “Tapestry: An infrastructure for fault-tolerant wide-area location and routing”, Berkeley, Computer Science Division, University of California, 2001